DOCUMENT RESUME

ED 108 466 FL 006 585

AUTHOR	Smith, Robert Lawrence, Jr.
TITLE	The Syntax and Semantics of ERICA. Technical Report No. 185, Psychology and Education Series.
INSTITUTION	Stanford Univ., Calif. Inst. for Mathematical Studies in Social Science.
PUB DATE	14 Jun 72
NOTE	328p.
EDRS PRICE	MF-$0.76 HC-$17.13 PLUS POSTAGE
DESCRIPTORS	Ambiguity; *Child Language; Computational Linguistics; Context Free Grammar; Descriptive Linguistics; Grammar; *Language Development; Language Patterns; Language Research; Lexicology; Linguistic Theory; *Mathematical Linguistics; Phrase Structure; Psycholinguistics; *Semantics; Structural Analysis; *Syntax

ABSTRACT

This report is a detailed empirical examination of Suppes' ideas about the syntax and semantics of natural language, and an attempt at supporting the proposal that model-theoretic semantics of the type first proposed by Tarski is a useful tool for understanding the semantics of natural language. Child speech was selected as the best place to find data on natural language because it presents a view of the real problems represented by natural language, and because it allows the study of the process of language development. The main body of data consists of a series of recordings between a 32-month-old girl, Erica, and several adults. The ERIC corpus is found to be syntactically simpler and semantically more straightforward than adult speech. It is divided into utterances; its vocabulary is compared to ADULT vocabulary; a word frequency count is made; and its words are classified gramatically. A discussion follows of the standard concepts and results of the theory of generative grammars. A grammar for ERICA is devised, with special attention to lexical ambiguity. Mathematical syntax and semantics are discussed, followed by a description of ERICA semantics, with special reference to grammatical and semantic ambiguity. Conclusions include: (1) a reasonable probabilistic grammar for ERICA can be constructed; (2) the grammar GE1 is the best model for lexical disambiguation; (3) the grammar functions reasonably well semantically; (4) the notion of probability can play a key role in the construction of a semantics; and (5) simple set-theoretical functions are often successful in describing the ERICA semantics. (Author/AM)

THE SYNTAX AND SEMANTICS OF ERICA

BY

ROBERT LAWRENCE SMITH, JR.

TECHNICAL REPORT NO. 185

JUNE 14, 1972

PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY
Robert Lawrence Smith, Jr.
TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER.

FL006585

PSYCHOLOGY & EDUCATION SERIES

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

TECHNICAL REPORTS

PSYCHOLOGY SERIES

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES

(Place of publication shown in parentheses; if published title is different from title of Technical Report, this is also shown in parentheses.)

(For reports no. 1 - 44, see Technical Report no. 125.)

50 R. C. Atkinson and R. C. Calfee. Mathematical learning theory. January 2, 1963. (In B. B. Wolman (Ed.), Scientific Psychology. New York: Basic Books, Inc., 1965. Pp. 254-275)

51 P. Suppes, E. Crothers, and R. Weir. Application of mathematical learning theory and linguistic analysis to vowel phoneme matching in Russian words. December 28, 1962.

52 R. C. Atkinson, R. Calfee, G. Sommer, W. Jeffrey and R. Shoemaker. A test of three models for stimulus compounding with children. January 29, 1963. (J. exp. Psychol., 1964, 67, 52-58)

53 E. Crothers. General Markov models for learning with inter-trial forgetting. April 8, 1963.

54 J. L. Myers and R. C. Atkinson. Choice behavior and reward structure. May 24, 1963. (Journal math. Psychol., 1964, 1, 170-203)

55 R. E. Robinson. A set-theoretical approach to empirical meaningfulness of measurement statements. June 10, 1963.

56 E. Crothers, R. Weir and P. Palmer. The role of transcription in the learning of the orthographic representations of Russian sounds. June 17,. 1963.

57 P. Suppes. Problems of optimization in learning a list of simple items. July 22, 1963. (In Maynard W. Shelly, II and Glenn L. Bryan (Eds.), Human Judgments and Optimality. New York: Wiley. 1964. Pp. 116-126)

58 R. C. Atkinson and E. J. Crothers. Theoretical note: all-or-none learning and intertrial forgetting. July 24, 1963.

59 R. C. Calfee. Long-term behavior of rats under probabilistic reinforcement schedules. October 1, 1963.

60 R. C. Atkinson and E. J. Crothers. Tests of acquisition and retention, axioms for paired-associate learning. October 25, 1963. (A comparison of paired-associate learning models having different acquisition and retention axioms, J. math. Psychol., 1964, 1, 285-315)

61 W. J. McGill and J. Gibbon. The general-gamma distribution and reaction times. November 20, 1963. (J. math. Psychol., 1965, 2, 1-18)

62 M. F. Norman. Incremental learning on random trials. December 9, 1963. (J. math. Psychol., 1964, 1, 336-351)

63 P. Suppes. The development of mathematical concepts in children. February 25, 1964. (On the behavioral foundations of mathematical concepts. Monographs of the Society for Research in Child Development, 1965, 30, 60-96)

64 P. Suppes. Mathematical concept formation in children. April 10, 1964. (Amer. Psychologist, 1966, 21, 139-150)

65 R. C. Calfee, R. C. Atkinson, and T. Shelton, Jr. Mathematical models for verbal learning. August 21, 1964. (In N. Wiener and . P. Schoda (Eds.), Cybernetics of the Nervous System: Progress in Brain Research. Amsterdam, The Netherlands: Elsevier Publishing Co., 1965. Pp. 333-349)

66 L. Keller, M. Cole, C. J. Burke, and W. K. Estes. Paired associate learning with differential rewards. August 20, 1964. (Reward and information values of trial outcomes in paired associate learning. (Psychol. Monogr., 1965, 79, 1-21)

67 M. F. Norman. A probabilistic model for free-responding. December 14, 1964.

68 W. K. Estes and H. A. Taylor. Visual detection in relation to display size and redundancy of critical elements. January 25, 1965, Revised 7-1-65. (Perception and Psychophysics, 1966, 1, 9-16)

69 P. Suppes and J. Donio. Foundations of stimulus-sampling theory for continuous-time processes. February 9, 1965. (J. math. Psychol., 1967, 4, 202-225)

70 R. C. Atkinson and R. A. Kinchla. A learning model for forced-choice detection experiments. February 10, 1965. (Br. J. math stat. Psychol., 1965, 18, 184-206)

71 E. J. Crothers. Presentation orders for items from different categories. March 10, 1965.

72 P. Suppes, G. Groen, and M. Schlag-Rey. Some models for response latency in paired-associates learning. May 5, 1965. (J. math. Psychol., 1966, 3, 99-128)

73 M. V. Levine. The generalization function in the probability learning experiment. June 3, 1965.

74 D. Hansen and T. S. Rodgers. An exploration of psycholinguistic units in initial reading. July 6, 1965.

75 B. C. Arnold. A correlated urn-scheme for a continuum of responses. July 20, 1965.

76 C. Izawa and W. K. Estes. Reinforcement-test sequences in paired-associate learning. August 1, 1965. (Psychol. Reports, 1966, 18, 379-919)

77 S. L. Blehart. Pattern discrimination learning with Rhesus monkeys. September 1, 1965. (Psychol. Reports, 1966, 19, 311-324)

78 J. L. Phillips and R. C. Atkinson. The effects of display size on short-term memory. August 31, 1965.

79 R. C. Atkinson and R. M. Shiffrin. Mathematical models for memory and learning. September 20, 1965.

80 P. Suppes. The psychological foundations of mathematics. October 25, 1965. (Colloques Internationaux du Centre National de la Recherche Scientifique. Editions du Centre National de la Recherche Scientifique. Paris: 1967. Pp. 213-242)

81 P. Suppes. Computer-assisted instruction in the schools: potentialities, problems, prospects. October 29, 1965.

82 R. A. Kinchla, J. Townsend, J. Yellott, Jr., and R. C. Atkinson. Influence of correlated visual cues on auditory signal detection. November 2, 1965. (Perception and Psychophysics, 1966, 1, 67-73)

83 P. Suppes, M. Jerman, and G. Groen. Arithmetic drills and review on a computer-based teletype. November 5, 1965. (Arithmetic Teacher, April 1966, 303-309.

84 P. Suppes and L. Hyman. Concept learning with non-verbal geometrical stimuli. November 15, 1965.

85 P. Holland. A variation on the minimum chi-square test. (J. math. Psychol., 1967, 3, 377-413).

86 P. Suppes. Accelerated program in elementary-school mathematics -- the second year. November 22, 1965. (Psychology in the Schools, 1966, 3, 294-307)

87 P. Lorenzen and F. Binford. Logic as a dialogical game. November 29, 1965.

88 L. Keller, W. J. Thomson, J. R. Tweedy, and R. C. Atkinson. The effects of reinforcement interval on the acquisition of paired-associate responses. December 10, 1965. (J. exp. Psychol., 1967, 73, 268-277)

89 J. I. Yellott, Jr. Some effects on noncontingent success in human probability learning. December 15, 1965.

90 P. Suppes and G. Groen. Some counting models for first-grade performance data on simple addition facts. January 14, 1966. (In J. M. Scandura (Ed.), Research in Mathematics Education. Washington, D. C.: NCTM, 1967. Pp. 35-43.

91 P. Suppes. Information processing and choice behavior. January 31, 1966.

92 G. Groen and R. C. Atkinson. Models for optimizing the learning process. February 11, 1966. (Psychol. Bulletin, 1966, 66, 309-320)

93 R. C. Atkinson and D. Hansen. Computer-assisted instruction in initial reading: Stanford project. March 17, 1966. (Reading Research Quarterly, 1966, 2, 5-25)

94 P. Suppes. Probabilistic inference and the concept of total evidence. March 23, 1966. (In J. Hintikka and P. Suppes (Eds.), Aspects of Inductive Logic. Amsterdam: North-Holland Publishing Co., 1966. Pp. 49-65.

95 P. Suppes. The axiomatic method in high-school mathematics. April 12, 1966. (The Role of Axiomatics and Problem Solving in Mathematics. The Conference Board of the Mathematical Sciences, Washington, D. C. Ginn and Co., 1966. Pp. 69-76.

(Continued on inside back cover)

ED108466

THE SYNTAX AND SEMANTICS OF ERICA

by

Robert Lawrence Smith, Jr.

TECHNICAL REPORT NO. 185

June 14, 1972

PSYCHOLOGY AND EDUCATION SERIES

Reproduction in Whole or in Part Is Permitted for
Any Purpose of the United States Government

Copyright © 1972 by
Robert Lawrence Smith, Jr.

INSTITUTE FOR MATHEMATICAL STUDIES IN THE SOCIAL SCIENCES
STANFORD UNIVERSITY
STANFORD, CALIFORNIA

PREFACE

This work is a detailed empirical examination of Professor Patrick Suppes' ideas about the syntax and semantics of natural language. Readers familiar with his work will recognize the debt that I owe to Professor Suppes.

Several persons deserve special mention. The members of my committee: Julius Moravcsik, John McCarthy, and Dov Gabbay; for collecting the ERICA corpus: Arlene Moskowitz; for his superb understanding of computer science: David Levine; for their assistance in statistics: Mario Zanotti and Charles Dunbar; for editing: Dianne Kanerva and Florence Yager; for reading the complete text: Edward Bolton; for the most detailed and patient assistance I received: my wife, Nancy Smith.

I would also like to thank the following good people for their assistance at many points and in many different ways: Barbara Anderson, Naomi Baron, Marnie Beard, Lee Blaine, Alex Cannara, Phyllis Cole, Clark Crane, Kathleen Doyle, Dexter Fletcher, Jamesine Friend, Betsy Gammon, Adele Goldberg, Pentti Kanerva, Joanne Leslie, Buddy Mancha, Lillian O'Toole, Ron Roberts, Marguerite Shaw, Rainer Schulz, Steve Weyer, Robert Winn.

The entire dissertation was done on the IMSSS PDP-10 and the Stanford AI PDP-10, mostly at IMSSS. As a result, the format is somewhat different from dissertations typed on a conventional typewriter. Linear notation is used throughout. Exponentiation is indicated by the symbol '^' as in

x^2

which is read "x square". References to footnotes occur on the line, rather than above, as is customary. In some chapters (especially 6), the format is a bit unusual. These inconveniences are, I believe, offset by the fact that performing this research and reporting on it in any detail is almost impossible without the computer.

Partial support for the research presented in this dissertation was supplied by the National Science Foundation under grant NSF-GJ443X.

TABLE OF CONTENTS

Section Page

CHAPTER 1 -- INTRODUCTION 1
- I. THE EXPERIMENT 1
- II. BACKGROUND — PREVIOUS WORK 2
- III. THE APPROACH TO THE DATA 9
- IV. TOWARDS A COMPUTER-PERFORMANCE THEORY OF AMBIGUITY 10
- V. METHODOLOGY AND ASSUMPTIONS 13
- VI. CONCLUSIONS 14

CHAPTER 2 -- THE ERICA CORPUS 17
- I. THE SELECTION OF A CORPUS 17
- II. SUPERFICIAL SYNTACTICAL FEATURES 20
- III. UTTERANCES: NOTATION AND CONVENTIONS 23
- IV. COMPARISON OF ERICA AND ADULT VOCABULARIES 26
- V. IMITATION OF WORD USAGES 32
- VI. COMPARISON OF THE CORPUS VOCABULARY TO THE VOCABULARY OF WRITTEN ENGLISH 34
- VII. DICTIONARY CONSTRUCTION 39
- VIII. WORD CLASSIFICATIONS 44
- IX. GOODNESS-OF-FIT TESTS ON THE ERICA AND ADULT DICTIONARIES 51

CHAPTER 3 -- FORMAL DEVELOPMENTS 56
- I. GENERATIVE GRAMMARS 56
- II. THE RELATION OF GENERATIVE GRAMMARS TO AUTOMATA 62
- III. DERIVATIONS AND TREES 63
- IV. CHOMSKY NORMAL FORM GRAMMARS 66
- V. LEXICAL SIMPLIFICATION OF CONTEXT-FREE GRAMMARS 66

CHAPTER 4 — A GRAMMAR FOR ERICA 70
- I. THE SIMPLE MODEL 70
- II. PROBABILITY AND LINGUISTICS 72
- III. MAXIMUM LIKELIHOOD AND ESTIMATIONS 85
- IV. CHI-SQUARE AND GOODNESS OF FIT TESTS 89
- V. GEOMETRIC MODELS FOR CFG 92
- VI. LEXICAL AMBIGUITY AND PROBABILISTIC GRAMMARS 94
- VII. THE GRAMMAR GE1 105
- VIII. LEXICAL AMBIGUITY IN THE ERICA CORPUS 108
- IX. PROBABILISTIC GRAMMARS AND UTTERANCE LENGTH 128

CHAPTER 5 — SEMANTICS 131
I. METAMATHEMATICAL SYNTAX AND SEMANTICS 131
II. CONTEXT-FREE AND METAMATHEMATICAL SYNTAX 136
III. MODEL STRUCTURES AND CFG 150
IV. SEMANTICS FOR ERICA 157
V. SEMANTICS FOR GE1 170

CHAPTER 6 -- THE SEMANTICS OF ERICA 198
I. THE SEMANTICS OF THE GRAMMAR GE1 198
1. ADJECTIVE PHRASE RULES 200
2. ADVERBIAL PHRASE RULES 205
3. QUANTIFIER-ARTICLE RULES 205
4. ADJECTIVE PHRASE RULES — POSSESSIVE ADJECTIVES 207
5. RULES FOR ADJECTIVE-PHRASES NOT PRECEDING NOUN PHRASES 209
6. RULES INTRODUCING POSSESSIVES 213
7. NOUN-PHRASE RULES 214
8. VERB-PHRASE RULES 222
9, RULES FOR NOUN-PHRASES THAT STAND ALONE 254
10. RULES GENERATING SENTENCES 257
11. PREPOSITIONAL PHRASE GENERATION 290
12. SUBJECTS OF SENTENCES 290
13. UTTERANCE-GENERATING RULES 292
II. GRAMMATICAL AND SEMANTICAL AMBIGUITY 302
III. PROBABILISTIC DISAMBIGUATION 310

BIBLIOGRAPHY 313

INDEX 315

(Appendices 1-7 are not included in this report.)

CHAPTER 1 -- INTRODUCTION

I. THE EXPERIMENT

My purpose in this work is to add weight to the proposal that model-theoretic semantics of the type first proposed by Tarski (1) is a useful tool for understanding the semantics of natural languages. This approach has been considered in very sophisticated ways (2); but it is seldom that a discussion of model-theoretic semantics has centered around a corpus of spoken or written English actually gathered under empirically sound conditions (3).

My first aim is to lay out such an experiment. I have completed the editing of a series of recordings between a 32-month-old child (Erica by name) and several adults. An extended description of this corpus is given in Chapter 2. To manage this corpus, which runs several hundred pages, I have transcribed the text onto the PDP-10 timesharing system at the Computer Based Laboratory of the Institute for Mathematical Studies in the Social Sciences, and I have written a number of programs to assist in the analysis.

---

(1) Alfred Tarski, "The Concept of Truth in Formalized Languages". in Logic, Semantics, and Metamathematics, London, 1955.

(2) See, for example, the series of papers by Richard Montague, some of which are listed in the Bibliography to this work.

(3) See, for example, the articles by Patrick Suppes and Elizabeth Gammon listed in the Bibliography of this work.

The use of the computer is an essential part of this work. In the beginning, the computer was used solely as a bookkeeper for the detail I could not manage alone, but as the analysis progressed the computer played a conceptually more important role.

II. BACKGROUND -- PREVIOUS WORK

Set-theoretical semantics is a standard way of discussing the meaning of the formal languages of mathematical logic. The standard body of results known as model-theory leaves little doubt as to the power of this method, whereby such historically important concepts as entailment, inference, truth, tense, and modality are opened to scientific examination in a comprehensive way. The major problem of relating these results to the questions surrounding the semantics of natural languages involves the characterization of the syntax of natural language in a way that relates it to the proposed semantics.

A. ENGLISH AS A FORMAL LANGUAGE -- MONTAGUE

Let me briefly review here the important work of Professor Richard Montague in connection with the semantics of natural languages (4). Montague bases his syntax of English on the notion of grammatical category in a system similar to the categorial grammars of Polish logicians of the 1930's (5). The semantics is then based on a tensed intensional logic--an artificial language designed for the perspicacity of its semantics. Montague gives several examples of English sentences, shows their translations into his artificial language, and discusses the semantic results as related to problems of intension, modality, and quantification.

Montague raises an important issue with his treatment of ambiguity. He remarks that a sentence can have two or more different semantic interpretations, and that these interpretations can correspond to alternative informal analyses. Several sentences are offered that have different semantic interpretations corresponding to de dicto and de re modalities. An example of this kind of modal ambiguity is the sentence:

---

(4) Specifically I will discuss the article: Richard Montague, "The Proper Treatment of Quantification in Ordinary English", forthcoming in Approacnes to Natural Language, J. Hintikka, J. Moravcsik, and P. Suppes, (editors), Dordrecht, Holland.

(5) Montague cites K. Ajudukiewicz, Jezyki Poznanie, Warsaw, 1960, as a source for nis work.

John seeks a unicorn.

Implicit in his remarks is the idea that competing philosophical views can be formally represented by alternative semantical interpretations.

More directly relevant to my work, several of Montague's sentences involve ambiguities resulting from other causes than modality. He notes that the sentence:

*) A woman loves every man.

can have two meanings, and follows through by showing that his semantics yields both of the following interpretations, here symbolized in my own notation.

1) $(\exists x)[\mathrm{WOMAN}(x) \wedge (\forall y)(\mathrm{MAN}(y) \rightarrow \mathrm{LOVE}(x,y))]$

2) $(\forall y)[\mathrm{MAN}(y) \rightarrow (\exists x)(\mathrm{WOMAN}(x) \wedge \mathrm{LOVE}(x,y))]$.

Montague does not reject alternative semantic interpretations as being spurious. Unfortunately, he has no theory for handling them either.

B. PROBABILISTIC GRAMMARS -- SUPPES AND GAMMON

My work is closely related to the work of Professor Patrick Suppes and his student Dr. Elizabeth Gammon, so I will discuss their contributions briefly here, and in more detail in the later chapters.

In "Probabilistic Grammars for Natural Languages" (6), Suppes assigns probabilities to the production rules of a phrase-structure grammar, and suggests that such grammars be used in describing the main features of a corpus of language--preferably a corpus recorded from actual speakers. Suppes explains:

> The probabilistic program ... is meant to be supplementary rather than competitive with traditional investigations of grammatical structure. The large and subtle linguistic literature on important features of natural language syntax constitutes an important and permanent body of material. ... one objective of a probabilistic grammar is to account for a high percentage of a corpus with a relatively simple grammar and to isolate the deviant cases that need additional analysis and explanation. At the present time, the main tendency in linguistics is to look at the deviant cases and and not to concentrate on a quantitative account of that part of a corpus that can be analyzed in relatively simple terms. (7)

Two important motives for Suppes' use of

---

(6) Patrick Suppes, "Probabilistic Grammars for Natural Languages", Technical Report No. 154, Institute for Mathematical Studies in the Social Sciences, Stanford, California.

---

(7) [Suppes-1], pp. 4-5.

probabilistic grammars are 1) determination of the central (syntactic) tendencies, and 2) isolation of (syntactic) problems for further study. These motives are also central in my work, but with semantics as the primary goal. As an example of the application of a probabilistic grammar, Suppes demonstrates the use of probabilistic grammars in the prediction of utterance length (8).

Suppes uses the noun-phrases from the ADAM-1 corpus of Roger Brown for the construction of probabilistic grammars (9). However, the ADAM-1 corpus is not sufficiently large or protracted for this kind of work.

Dr. Elizabeth Gammon continues the study of probabilistic grammars in a later paper (10) concerning the language of basal readers. The thrust of Gammon's work is the analysis of instructional materials; however, I have benefited from looking at the techniques she uses for classifying words into lexical categories and constructing grammars. Gammon also uses categorial grammars (similar to

---

(8) Patrick Suppes, "Semantics of Context-Free Fragments of Natural Languages", Technical Report No. 171, IMSSS, Stanford, Californial. See especially pp. 26-28.

(9) See [Suppes-1] and [Suppes-2].

(10) Elizabeth Macken Gammon, "A Syntactic Analysis of Some First-Grade Readers", Technical Report No. 155, Institute for Mathematical Studies in the Social Sciences, Stanford University.

Montague's syntax), so it is interesting to see the relative merits of generative grammars and categorial grammars. Context-free grammars have the advantage of being closer to current notation in linguistics; more deeply, context-free grammars allow the use of more parameters than the usual categorial grammars, so I consider only the use of context-free grammars.

Neither Suppes nor Gammon considers in any detail the problem of classifying words as to grammatical type, although both of them assume that this is done prior to the analysis. (Editors made the classifications for ADAM-1 and for Gammon's basal readers.) Montague considers only a few words ('walks','loves','ninety', 'temperature') and is not concerned with any empirical problems. I think that an empirical theory such as mine must consider the problem of dealing with several thousand words in a convenient way, particularly for computer implementation. Hence, I have used a dictionary to provide information about the grammatical functions that words can perform.

C. SEMANTICS OF CONTEXT-FREE LANGUAGES -- SUPPES

In his more recent work (11) Suppes has become primarily concerned with semantics. In "Semantics of Context-Free Fragments of Natural Languages", Suppes gives a context-free grammar for the noun-phrases in ADAM-1, and defines semantic functions on the rules of that grammar. Suppes emphasizes the use of simple semantic functions in as many cases as possible, attempting to isolate remaining difficulties.

---

(11) Patrick Suppes, "Semantics of Context-free Fragments of Natural Languages, Technical Report No. 171, IMSSS.

In the main, I have used Suppes' formulations for semantics rather than Montague's. (See Chapter 5 for my formulation.) Suppes bases his semantics on a context-free grammar and does not translate his English syntax into some artifical language prior to semantic analysis. These are advantages to his approach, I believe.

In considering alternative semantical functions for certain constructions, (mainly the "double noun" construction as in the phrases 'Daddy suitcase' and 'Baby Ursula'), Suppes also allows alternative semantic interpretations. Unfortunately, these alternative semantic interpretations do not in Suppes' system necessarily rest on alternative syntactic representations (or "trees"), as was the case in Montague's work.

There are two main problems involved here. First, it is my belief that syntax and semantics correspond very closely, so I would prefer to have a different syntactic structure to represent each semantic interpretation. In addition, any help that a probabilistic grammar may have in selecting between alternative semantic interpretations is obscured by having two or more semantic interpretations arise from one syntactic representation.

III. THE APPROACH TO THE DATA

In the context of previous work, the purpose of my work is to supply a detailed examination of a large corpus of data using mainly the methods of Professor Suppes, and to extend those methods where possible. In the case of Suppes' work on ADAM-1, the size of the corpus and the age of the child required Suppes to confine his analysis, in the main, to the noun-phrase fragment of ADAM-1. With the larger ERICA corpus, I have written a more complete utterance grammar and semantics. The size of the ERICA corpus (over 9,000 child utterances) has made this a large task of computation and data manipulation.

While Montague's work is not addressed to any empirical problems, nevertheless I believe that theoretical work similar to his can benefit from empirical work in two ways. First, there is a tendency in theoretical work to be confined to one's own small sample of sentences, and a danger of error if the only criterion of success is the force, largely psychological, of a few competing examples and counterexamples. Second, there is the chance that theoretically interesting examples may abound in empirical data. An example of this kind, I believe, is the beginning of an extension of the theory of definite descriptions that I have given in Chapter 5, based on the uses of the word 'the' in ERICA.

Theories of language have been labeled as being competence or performance theories. Admitting this terminology, my work is decidedly in the performance camp, although not with any hostility. In fact the two kinds of research are both important. I call the basic approach of this work "computer-performance". By this I mean that I am trying to describe linguistic behavior with a theory that is largely implementable on a computer. I am not really arguing the relative computational abilities of the computer and the human mind, or the nature of intelligence and how to develop it artificially. Rather, I am using the computer as a tool for formulating and testing a theory in an exact way.

IV. TOWARDS A COMPUTER-PERFORMANCE THEORY OF AMBIGUITY

I am trying to develop a methodology for linguistics research that will allow the comparison of conflicting philosophical/linguistic views in a scientifically acceptable way, building on the results in these areas, and bringing them into focus around a performance theory. Because of the pervasiveness of ambiguity in any theory of language, I have devoted a good part of this work to considering how to handle ambiguity.

I identify and distinguish several kinds of ambiguity in ERICA (as I refer to the corpus), which are:

1) Lexical ambiguity: ambiguity due to multiple entries in the dictionary (Chapters 2 and 4).

2) Grammatical ambiguity: ambiguity present syntactically in a grammar (Chapters 3 and 4).

3) Semantic ambiguity: two (or more) "meanings" for an utterance (Chapter 6).

I believe that many problems of the semantics of natural languages can be characterized as problems of ambiguity. I think that each utterance in English has only a small number of "plausible" semantic interpretations. The alternative is, I believe, to adjudge the human language processing facility as arbitrarily complex and inherently anomalous.

My analysis of the "plausible" is in probabilistic terms. Given the syntax provided by the probabilistic grammar, the obvious extension is to let the probability of a semantic interpretation be the probability of the syntactic structure(s) associated with that interpretation. (Two or more syntactic representations of a sentence may have the same semantic interpretation, I believe.)

The use of the probabilistic grammar in disambiguating provides an interesting check on the relation of the syntax to the semantics. We can ask, for a syntactic construction that has alternative semantic representations, if the probabilities associated with those interpretations correspond to our intuitions about the utterances in the corpus using the construction.

I use probabilistic grammars to disambiguate in two ways. First, there is in ERICA a certain amount of ambiguity due to the dictionary (lexical ambiguity). This kind of ambiguity is often only apparent and should be dismissed without further consideration. In Chapter 4 I discuss several ways to remove this lexical ambiguity. The most intuitively satisfactory method is to accept the alternative with the highest probability. Secondly, in a more detailed discussion in Chapter 6, I discuss the grammatical ambiguity (ambiguity due to the grammar rather than the dictionary) remaining in ERICA after all lexical ambiguity has been removed, and I conduct a careful examination of the success of probabilistic disambiguation on these cases.

Strictly interpreted, these results indicate mixed success. However, what they indicate to me are the many ways in which the dictionary and the grammar can be improved, and they suggest what features are causing the major difficulties.

V. METHODOLOGY AND ASSUMPTIONS

Let me summarize the basis of this work by listing what I attempted to do as METHODOLOGY and the justifications as ASSUMPTIONS.

A. METHODOLOGY

1) The data base (Erica's memory, semantic information) is characterized as a set-theoretical structure (Chapter 5). The lexicon greatly simplifies the kinds of things in this structure by classing words as nouns, verbs, and so on.

2) The syntax of the child's speech is generated by a context-free grammar, designed to remove most lexical ambiguities by rejecting most alternative interpretations. Remaining interpretations should represent genuine ambiguities. Further ambiguity is handled by the probabilistic nature of the grammar (which selects the "most likely" interpretation as a first approximation).

3) The meaning of an utterance is computed by set-theoretic functions into the 'objects' in the data base.

B. ASSUMPTIONS

1) The "deep structure" of the semantics likely

corresponds to the "surface structure" of the syntax, at least more than supposed.

2) The understanding of natural language is a phenomenon open to our understanding to the point that we can simulate it on a computing machine of reasonable size.

3) Much language processing is done in a syntactical way (albeit in a way that corresponds to the semantics.) Certain semi-automatic linguistic reflexes are learned in such a way that the full power of the semantic machinery is not needed.

4) One need not be concerned that obvious simplifications in the analysis (such as my handling of quantifiers, verbs, adverbs) will so grossly misrepresent the problem that the whole enterprise is valueless. This is more than an article of faith in that it corresponds to my feeling that speakers commonly simplify the semantic structure of concepts in many ordinary contexts. Quantifiers tend to look like simple adjectives, modal concepts such as 'necessity' are assumed to be transparent, and verbs look like simple 1-place predicates.

VI. CONCLUSIONS

I make the following conclusions from the work reported here. These results are readily classed into 'empirical' and 'conceptual' issues.

A. EMPIRICAL ISSUES

1) A reasonable probabilistic grammar for ERICA can be constructed. My grammar GE1 recognizes 77 percent of the ERICA corpus, removes most of the lexical ambiguity present in the corpus, and introduces very little grammatical ambiguity. (Chapters 4 and 6)

2) Further, the grammar GE1 can be used to complete the process of lexical disambiguation in an impressive way by selecting the most likely lexical alternative. This method is apparently better than the other models of lexical disambiguation that I suggest. (Chapter 4)

3) Semantically, the grammar functions reasonably well. Many rules are obviously correct. Many of the remaining problems can be ascribed to the need for a dictionary that more completely describes the alternative uses of words in the corpus, and to subtler rules. (In this first pass of the data, I simply used a dictionary and grammar constructed mostly a priori.) (Chapters 5 and 6)

B. CONCEPTUAL ISSUES

1) There is a need, philosophically, to study the performance side of linguistic concepts by looking at corpora of data. (See for example the discussion of the word 'the' in Chapter 5.)

2) There is a relation between the syntax of the

formal languages of mathematical logic and generative grammars. This relationship provides a practical and conceptual basis for the set-theoretical semantics of context-free languages. (Chapter 5)

3) There is a tradeoff between symbols that denote objects and symbols that call upon functions. This tradeoff has implications, I believe, both to certain philosophical disputations and to computer-based semantic systems. (Chapter 5)

4) A useful part of a theory of set-theoretical semantics can be the inclusion of one or more contextual parameters, indicating sets of objects currently under consideration in the conversation.

5) An extended theory of definite descriptions can be made, using contextual parameters, that accounts for the classical theory as well as the other observed uses of the word 'the'.

6) The notion of probability can play a key role in the construction of a semantics. This can be effected by probabilistic grammars.

7) Simple set-theoretical functions are often successful in describing the ERICA semantics. I have no single measure of correctness, but rather a detailed examination of the syntax rules and their associated semantic functions.

CHAPTER 2 -- THE ERICA CORPUS

I. THE SELECTION OF A CORPUS

Erica is a little girl. Arlene Moskowitz of Berkeley collected recordings of Erica talking to adults, usually to Arlene herself or to Erica's mother, but occasionally to Erica's father. At the beginning of the recording in 1969 Erica was 31 months old, and she was 33 months old at the end. (Erica was born on July 24, 1966. Unfortunately, the dates of all the recordings are not available.) The tapes were made in her family's apartment, where the surroundings were familiar to Erica. An effort was made to have normal conversation, and the impression from the transcriptions is that the awareness of the recording equipment was forgotten after the fourth or fifth tape. Most of the recordings were of a one-hour session, but some extended over several days, a few minutes each day. Miss Moskowitz began the editing but did not finish, so I cannot vouch for the authenticity of the data, except to say that I have tried to edit the text myself, and that I alone am responsible for any effect that remaining errors may have on my results (1).

Several reasons persuaded me that the speech of a child was the appropriate place to look for the data for this experiment; these reasons are discussed below.

1) There was reason to believe that children's speech was syntactically simpler than adult speech, and this has proven to be the case. Compared to the adult text in the ERICA corpus (giving a name to the corpus itself), Erica's utterances are shorter, the vocabulary less rich, and the structure is more repetitive. So, if by natural language we mean spoken, informal conversation, the speech of a child would be the natural candidate for a simple beginning.

2) I had hoped that Erica's speech would be more semantically straightforward compared to adult speech. I have no reason to doubt that this assumption is correct. Simple semantical functions appear to be successful in an encouraging part of Erica's speech. This was not surprising to me, since I expect semantical features of language to have their syntactic counterparts. The syntactical simplicity of child speech then suggests semantical simplicity.

3) The developmental aspects of language and concepts are philosophically interesting, and it is these

---

(1) I would like to thank Barbara Anderson, Robert Winn, and Florence Yager of the Institute staff for their help in typing the ERICA corpus into the PDP-10 computer for this analysis.

factors that one would most expect to find in the study of child language, particularly if the study were well timed and protracted, covering the first moments of speech well into nursery school. Since the ERICA corpus was collected sporadically and hastily (only two months from the first recording to the last), the possibility of studying language development in these particular data is remote.

Given that we want to look at the semantics of natural language, the question of the selection of a corpus bears some discussion. The advantage in selecting child language is that in it we are seeing something like the real problems that natural language represents, in rougnly the right mixtures. It certainly would impress no one to prove that model-theoretic semantics was useful for a patently artificial language, say ALGOL-60. Moreover, esoteric counterexamples to a model-theoretic approach would not impress me as being reason to abandon the project. What is needed is a detailed discussion of some genuine data.

The price paid for this spontaneity is that the data base for the meaning of the child's utterances is constantly shifting and impossible to separate, even for a moment of reflection, from such problems as perception and memory. The child's conversations free-wheel as quickly as the duration of attention span. The only recourse is to back away from the individual utterances and their inscrutable contexts and look for patterns that are more readily studied in classes of utterances.

In retrospect, looking at a corpus of free conversation is valuable for getting a feel for the kinds of grammars and semantic functions that are best. The real test should be conducted in a situation where the discussion can be limited in content. One solution might be to organize an experiment where children are encouraged to talk about certain fixed subjects, such as facts about baseball, or the objects strewn about the interviewing room. Another solution might be to look at spoken or written language concerning some precise subject matter such as elementary mathematics.

II. SUPERFICIAL SYNTACTICAL FEATURES

The most striking and permanent feature of the corpus is its size. There are 19,826 utterances in all, excluding utterances that were completely unidentifiable during transcription, but including utterances that could be partially understood. I used the symbol

<xxx>

to indicate unintelligibility of all or part of an utterance. Thus,

Can you <xxx>.

would be included as an utterance of length three. Using a similar notation,

<n> , <v> , <adj>

stand respectively for noun, verb, and adjective, when the exact word was not identifiable, but the editor thought she had good reason for a grammatical classification. The analysis of the length of utterances in this chapter first eliminated the utterances that included the unintelligibility symbol <xxx> since it might be standing for a whole phrase that was garbled on the tape.

Comments were included occasionally in the text when the editor believed that what she heard on the tape was not fully described by the utterances themselves; also comments about the situation leading up to the recording session itself were included. Of course, comments were not included in any syntactic study, and the comments were not sufficiently regular to admit any organized use in the semantic analysis, although I have noted the comments in the course of reading the corpus.

The text was prepared by the straightforward approach of trying to make a consistent and accurate copy of a conversation. It may be argued that a special representation, such as a phonetic system, would be more appropriate. I have no reason to really think so at this time, especially considering the problems that devising and using such a system would create. Phonetic representations were of course developed to capture the subtleties of sound. While I did not use a phonetic approach, it is clearly desirable from a semantic point of view. For example, the sentence

here it is

(unpunctuated!) can be either a question, a declaration, or an exclamation depending on the emphasis and the raising and lowering of the voice; these features are lost to my analysis.

A full implementation of a theory of language on a computer would of course include a system for recognizing spoken English and translating it into some kind of normal form. I assume that this translation would very much resemble written English, and it is for this reason that I defend the way ERICA was edited. If this assumption fails then some different representation of spoken English would have to be found.

III. UTTERANCES: NOTATION AND CONVENTIONS

The text is divided into utterances. If I were pressed to name an objective criterion for making the division between one utterance and the next I would suggest time-lag between sounds. However, it is clear from listening to the tapes that the editor has followed the interaction "semantically" and is trying to unitize the speech. That this is a natural process is indicated by the fact that the transcription is little different from other transcriptions of spoken English.

The units of speech seem to be rather like the "complete thoughts" of classical grammar. However formally elusive this idea may be I am drawn to it by looking at ERICA and comparing the divisions to what I imagine the conversation to have been like as an interaction.

Once the transcription is complete it is easy to define the delimitation of words in the utterances.

Notation: A *word* is an unbroken string of the characters

a,b,c, ... z,0,1, ... ,9,<,>,#,$,',-

occurring in an utterance. Lower and upper case letters are considered equivalent. The *length* of an utterance is the number of words in it.

Several characters are taken as having special significance.

1) The apostrophe ' is a part of words, as in possessives and contractions. In the case of contractions, the standard interpretation is taken formally in that we treat the contraction as though it were two dictionary words. However, a contraction only adds one to the length of an utterance. This has the advantage of treating the contraction in a way consistent with standard usage. The price paid is that I lose a possible correspondence between syntactical and semantical features of the utterance by having one word stand for perhaps two semantical "units".

EXAMPLES OF USES OF THE APOSTROPHE

WORD	MEANING
Erica's	the possessive of Erica
doesn't	the contraction of a verb and a negating particle
men's	the possesive of men

2) The dash - is a part of words, as in

ring-around-the-rosy

which is counted as one word.

3) The question mark ? denotes questions.

4) Quotes " (but not single quotes, which are not used due to the ambiguity with the apostrophe) indicate quotations and use-mention distinctions. I am not concerned with analyzing the semantics of these.

In standard English, punctuation characters (such as commas and semicolons) often indicate phrasing in sentences. I have not used these clues in the analysis formally, but it could be done by including punctuation characters as symbols generated by the grammar. Obviously punctuation is needed as phrase markings at some level in the analysis of natural language. Here I simply ignore punctuation altogether.

Of the utterances in the corpus, ERICA had 8,919 utterances with a mean length of 3.087, and ADULT had 10,695 utterances with a mean length of 4.838, excluding any utterances that were in part unintelligible. (The disparity between these numbers and the original counts of 9,085 and 10,740 reflects the number of partly unintelligible utterances.) A more complete analysis of the lengths of utterances in the corpus is included as Appendix 1.

IV. COMPARISON OF ERICA AND ADULT VOCABULARIES

Using the familiar type-token distinction, the ERICA corpus has 79,770 word tokens and 3,169 types. This count includes the symbols for unrecognized words, such as <n> used for a noun and <xxx> used for an unclassifiable word, but does not include utterances that were completely unintelligible. ERICA (the child's speech in the complete ERICA corpus) has 27,922 tokens and 1,853 types; ADULT (the adults' portion) has 51,848 tokens and 2,867 types. Appendices 2 and 3 list the words in ERICA and ADULT by rank and alphabetical ordering.

Obviously ERICA and ADULT have different vocabularies, and neither one uses all the words found in the other. However, it is of some interest to ask how different these vocabularies are and to propose measures of the difference. A simple test is to ask how many words occur in one but not the other. Of the words in ERICA, 301 types were not represented in ADULT. This comparison gives a misleading impression of the difference between the two vocabularies, since these 301 types account for only 565 tokens out of the 27,922 tokens in the ERICA vocabulary. The top 135 words in ERICA are all represented in ADULT, and most of the words in ERICA not found in ADULT have a small frequency, many occurring only once or twice.

If we look at the portions of the vocabularies with frequency greater than or equal to 5 we get a better impression of the similarity. There are 607 types in the ERICA vocabulary with frequency greater than or equal to 5, accounting for 25,678 tokens. Out of these, only 14 types, for 159 tokens, are not to be found in the ADULT vocabulary. Tables 1 and 2 summarize these results. Table 3 lists the words with frequency greater than or equal to 5 from ERICA not in ADULT at all, and Table 4 lists the words found in ADULT (freq >= 5) but not found in ERICA. (The string '>=' is read 'greater than or equal to'. Its use here reflects the fact that this work is being composed on the PDP-10 computer, and the the use of '>=' is standard linear notation.)

TABLE 1

WORDS IN THE ERICA VOCABULARY NOT FOUND IN THE ADULT VOCABULARY

Complete ERICA Vocabulary

	Types	Tokens
Size of sample	1,853	27,922
Words in ERICA not in ADULT	301	565
Percent not found	16.24%	2.02%

Portion of ERICA Vocabulary with Frequency >= 5

Size of sample	607	25,678
Words in ERICA not in ADULT	14	159
Percent not found	2.31%	.62%

TABLE 2

WORDS IN THE ADULT VOCABULARY NOT FOUND IN THE ERICA VOCABULARY

Complete ADULT Vocabulary

	Types	Tokens
Size of sample	2,867	51,848
Words in ADULT not in ERICA	1,315	2,861
Percent not found	45.87%	5.52%

Portion of ADULT Vocabulary with Frequency >= 5

	Types	Tokens
Size of sample	945	48,485
Words in ADULT not in ERICA	106	1,067
Percent not found	11.22%	2.20%

TABLE 3

WORDS OCCURRING IN ERICA VOCABULARY NOT IN ADULT VOCABULARY

(Frequency >=5)

Freq	Word
34	wanna
31	yup
16	lookat
13	momma
10	present
7	eek eh tap yeh
6	gobble luminum
5	grapefruits mouses sweetie

TABLE 4

WORDS OCCURRING IN ADULT VOCABULARY NOT IN ERICA VOCABULARY

(Frequency >= 5)

Freq	Word
84	else
77	were
37	things
30	which
28	understand
26	looks
23	much
20	breakfast sure
18	correct really
16	yourself
13	certainly few
12	building delicious feet real
11	already envelope song than
10	behind humm sorry until
9	count ears instrument minutes page tweet
8	boom closet ever everybody phone sat taste thought tired told wow
7	ate basket best cannot chickens each feed fireplace goodness happens lean lid lie line living meadow mind push squares whisper you'll
6	chinese comfortable its kitties lake lovely nail once party poor rhyme set toby
5	add ago anything apart bedroom different dinosaur dolly's fact growing haven't indians instruments loudly movie names park peck purr puts quite row rug sewing special stream television tooth you've

Some tentative conclusions are:

1. The ERICA and ADULT vocabularies are similar, especially at the high-frequency ends of the distributions. The bulk of their speech comes from the 1,552 words that are common to both lists. Erica draws 97.98 percent of her speech from the common vocabulary, and the adults 94.48 percent.

2. The ADULT vocabulary is more nearly a superset of the ERICA vocabulary than conversely. This holds throughout Tables 1 and 2. For example, only 16.24 percent of the words in ERICA do not occur in ADULT, while 45.87 percent of the words in ADULT do not occur in ERICA.

V. IMITATION OF WORD USAGES

A reasonable hypothesis about the speech of a child is that there is a strong tendency for the child to use words recently used by the an adult. As a simple test of this hypothesis, let a usage of a given word be an n-imitation occurrence if the word occurs in the previous n adult utterances. Table 5 gives the results of looking for n-imitations, n=1,2,...8, on the twenty hours of the ERICA corpus. To avoid confusing the comparisons, no counting was done until 8 adult utterances were found at the beginning of each hour.

TABLE 5

ERICA WORD USAGES THAT IMITATE ADULT WORD USAGES

(FIRST 8 ADULT UTTERANCES IN EACH HOUR ARE IGNORED)

N	N-IMITATION	NON-IMITATION	% IMITATION
1	3424	24498	12.26
2	4939	22983	17.69
3	5932	21990	21.24
4	6729	21193	24.01
5	7386	20536	26.45
6	7929	19993	28.40
7	8415	19507	30.14
8	8816	19106	31.57

Word Types = 3,169 (complete corpus)
Word Tokens = 79,770 (complete corpus)
ERICA Tokens = 27,922
ADULT Tokens = 51,848

VI. COMPARISON OF THE CORPUS VOCABULARY TO THE VOCABULARY OF WRITTEN ENGLISH

A standard computational analysis of written English texts is contained in Computational Analysis of Present Day American English by Henry Kučera and W. Nelson Francis (2). I want to compare the ERICA vocabulary to the vocabulary for the [K-F] corpus of written speech. There were 50,406 types in [K-F], representing 1,014,232 tokens. The samples comprising the [K-F] were selected to be a cross-section of contemporary American written English.

I have taken the 100 most common words in ERICA, looked up their frequencies in [K-F], and then used the [K-F] frequencies as the basis for the theoretical frequencies of a chi-square test. I summed up the frequencies for the 100 most frequent words in ERICA and [K-F], and called these sums the OBSERVED-SUM and the EXPECTED-SUM, respectively. The EXPECTED-FREQUENCY of a given word was then the word's frequency in [K-F] multiplied by

$$\frac{\text{OBSERVED-SUM}}{\text{EXPECTED-SUM}} .$$

---

(2) Brown University Press, 1967. Referred to as [K-F].

The chi-square contribution of the given word was then computed by the usual formula

$$\frac{(\text{OBSERVED-FREQUENCY} - \text{EXPECTED-FREQUENCY})^{2}}{\text{EXPECTED-FREQUENCY}} .$$

The results of this test are in Table 6. The indication is that Erica's speech is rather different from written English, even in terms of high-frequency words.

TABLE 6

GOODNESS-OF-FIT TEST

FREQUENCIES FOR THE FIRST 100 WORDS IN ERICA ESTIMATED BY [K-F]

(RELATIVIZED)

RANK	WORD	OBSERVED	REL.EXPECTED	CHI^2
1	you	3120	443.0232	16175.6860
2	a	2390	3132.8456	176.1401
3	the	2220	8098.8582	4267.3884
4	i	2178	697.4313	3143.0822
5	that	1775	1428.4331	84.0842
6	is	1728	1361.5616	98.6199
7	it	1716	1180.4965	242.9182
8	what	1692	257.2393	8002.4256
9	to	1439	3525.4456	1234.6099
10	and	1206	3889.8680	1851.7716
11	he	982	1286.6009	72.1138
12	are	948	592.2706	213.6582
13	do	942	183.7616	3128.6483
14	in	906	2877.2242	1350.5116
15	don't	895	65.9277	10425.9840
16	no	888	296.7420	1178.0809
17	that's	883	25.0768	29351.1390
*18	uh	836	.8089	862306.1800
19	on	786	908.9661	16.6350
20	this	717	693.7911	.7764
21	know	687	92.0830	3843.5547
*22	huh	675	.6741	674544.6900
23	have	650	531.3313	26.5037
24	go	630	84.3982	3527.1042
25	there	599	367.2536	146.2379
26	your	590	124.4402	1741.7681
27	we	572	357.6813	128.4175
28	did	543	140.7536	1149.5423

---

* Indicates words that seem special to the ERICA corpus. Some of these are not peculiar to ERICA but rather are seldom found in written English.

+ Indicates words that were spelled differently in [K-F] than in ERICA. For example, ERICA uses 'ok', but the preferred English is 'okay'.

29	what's	527	7.1455	37820.6290
30	me	516	159.2241	799.4332
31	can	506	238.9036	298.6163
32	yes	490	13.4143	11406.5960
*33	oh	485	16.0438	13707.5160
34	see	468	104.0821	1272.4200
35	one	458	443.8322	.4523
36	going	452	53.7938	2947.7070
37	here	441	101.1161	1142.4601
38	get	430	101.1161	1069.7076
39	they	428	487.7839	7.3273
40	want	422	44.3563	3215.2132
41	of	409	4908.9832	4125.0596
42	my	399	177.8295	275.0748
43	all	393	404.5991	.3325
44	up	386	255.4866	66.6718
45	for	371	1279.3206	644.9097
46	will	370	302.5393	15.0425
47	not	368	621.3920	103.3285
48	she	353	385.4545	2.7326
49	where	350	126.4625	395.1291
50	put	336	58.9170	1303.1053
*+51	ok	334	2.6964	40706.4570
52	those	319	114.5982	364.5789
53	it's	313	40.7161	1820.8666
54	very	299	107.3179	342.3665
55	with	296	982.7134	479.8706
56	little	293	112.0366	292.2951
57	right	290	82.6455	520.2444
58	like	283	173.9197	68.4139
59	some	279	218.0063	17.0648
60	now	272	177.1554	50.7775
61	there's	267	14.6955	4331.7605
62	doing	244	21.9759	2243.1263
63	them	241	241.1955	.0002
64	at	237	725.0697	328.5367
*65	mommy	236	.1348	412637.5400
66	make	226	107.0482	132.1790
67	be	217	859.7563	480.5264
68	does	215	65.3884	342.3181
69	out	208	282.5657	19.6862
70	big	207	48.5357	517.3701
71	who	207	303.6179	30.7459
72	her	206	409.4527	101.0935
73	look	202	53.7938	408.3205
74	eat	200	8.2241	4471.9739
75	was	200	1323.4072	953.6322
*76	daddy	185	.5393	63094.1140
77	say	182	67.9500	191.4261
78	think	181	58.3777	257.5682

79	good	174	108.8009	39.0707
80	him	174	487.9188	201.9701
81	he's	171	16.8527	1409.9478
82	down	168	120.6652	18.5686
83	his	166	943.3456	640.5565
*+84	uhuh	163	.8089	32519.4880
85	just	162	117.5643	16.7953
86	baby	158	8.3589	2678.8660
87	let	151	51.7714	190.1881
88	didn't	150	54.0634	170.2415
89	come	149	84.9375	48.3179
90	has	149	328.8295	98.3447
91	isn't	148	13.0777	1391.9927
92	you're	148	20.3580	800.2968
93	house	147	79.6795	56.8786
*94	lookit	144	.4045	50980.2180
95	would	143	365.9054	135.7914
96	more	142	298.7643	62.2556
97	book	130	26.0205	415.5075
98	girl	128	29.6607	326.0412
*99	gonna	128	2.1571	7341.3689
100	tape	128	4.7188	3220.8247

The only word in the first 100 words in ERICA not occurring in [K-F] at all was the word 'Erica', so actually this list goes to rank 101 from the original list. A number of words, especially proper nouns, seem special to ERICA, and these words (starred in Table 6) contribute the bulk of the enormous chi-square sum of 2,347,036. Striking these special words from the data, and recalculating, yields a chi-square sum of 208,000. This is still unacceptable, but it indicates that it may be possible to isolate some of the differences between written and spoken English. For example, some of the difficult chi-square contributions in the second run come from the high frequencies of contractions in ERICA. The word 'what's' contributes about 40,000, and 'that's' contributes some 31,000 to the 208,000 chi-square for the second run; these two words are the most generous contributors.

VII. DICTIONARY CONSTRUCTION

A conceptually important fact about the syntactic study undertaken in this work is that words were put into grammatical categories apart from the contexts in which they arose. This differs from the technique used by Elizabeth Gammon in her study of basal readers (3). Dr. Gammon looked at each sentence individually, and gave each sentence a "sentence type" based on how it appeared that the words functioned in that sentence. Of course, given words may well be used differently from sentence to sentence, and this occurred in Gammon's work.

When a word functions differently in different sentences, I call the word lexically ambiguous. This phenomenon is illustrated by the sentences:

1) There is snow on the ground
2) It will snow tomorrow.

According to the usual grammatical categories, the word 'snow' is a noun in 1) and a verb in 2).

The real difficulty with classifying the words individually in each sentence, as Gammon did, is that it leaves unanalyzed the crucial task of how one knows when a word is performing one syntactic function and not another. Lexical ambiguity is very widespread if one takes as a measure the number of multiple listings that words have in standard dictionaries. A theory of language must begin to account for the ubiquitous ambiguity of natural language in some way that makes it more than merely tiresome.

My partial solution is to create a dictionary for ERICA with multiple listings for a good portion of the words. In doing so I have not included all of the

---

(3) A Syntactic Study of First-Grade Readers, by Elizabeth Macken Gammon, Technical Report No. 155, June 22, 1970, IMSSS.

possibilities, or even all the ones that are probably represented in ERICA. To have done so would have obscured the results. The point is to implement in some detail a theory of lexical ambiguity, and to show how it might work in many cases, without letting the details become burdensome. With 78,000 word occurrences in ERICA, every occurrence of every word cannot be examined readily.

NOTATION: In the dictionary, each word is associated with a grammatical classification string. This string may be one classification; e.g., 'n' stands for noun in the dictionary. Or the classification string may be several classifications separated by commas. 'n,v' would be used for a word that could be either a noun or a verb.

Sometimes words (i.e., strings of word characters) are contractions. The pedestrian view is that contractions are two or more words that have been run together. For example, 'you' is a personal pronoun, and hence has the classification 'persp'. Supposing 'have' is a verb, it would have the classification 'v'. The word 'you've' is the contraction of 'you' with 'have' and has the classification 'persp#v'. (The symbol '#' stands for a space in the classification.) This notation merely says that 'you've' is to be thought of as 'you have'. The situation is, however, complicated by the fact that 'have' can be either a verb ("to possess") or an auxiliary verb and is thus classified 'v,aux'. This means that 'you've' can be 1) a personal pronoun followed by a verb, or 2) a personal pronoun followed by an auxiliary. The correct classification is therefore 'persp#v,persp#aux'.

To illustrate this in a sentence, consider:

*) You've seen him today.

Looking at the relevant portion of the dictionary:

WORD	GRAMMATICAL CLASSIFICATION	
him	persp	
seen	v	
today	adv	(adv is the symbol for adverb)
you've	persp#v,persp#aux	

Using a program written for the task, I look up the classifications and obtain

1) persp#v,persp#aux v persp adv

as the ambiguous lexical form for *). The ambiguous lexical form 1) is shorthand for saying that *) is either

2) persp v v persp adv

or

3) persp aux v persp adv .

The strings 2) and 3) are called alternative terminal forms for *). If the lexical form has only one alternative form, then I shall call it the terminal form.

The phrase 'terminal form' thus refers not to the original utterance but rather to the result of replacing the words in the utterance by their respective grammatical classifications in the dictionary. The Gammon method would have classed *) as 3), thus bypassing the lexical ambiguity that allows 2) as an alternative.

Dr. Gammon has told me privately she assumes that every utterance has a single terminal form, or at least a best one given the context of its use. While this assumption is useful, it is unsettling to me to leave the determination of the "best" terminal-form as a part of the given upon which a linguistic experiment rests. In Chapters 4 and 6 I try to resolve the natural ambiguities that arise from using the same words in different ways, so to a certain extent I am trying to use this assumption. Even so, Gammon's assumption is entirely too simple. It assumes that ambiguities are only apparent, that an adequate theory would always make a single selection. When I have laid out the necessary formal details, I shall try to argue that ambiguity plays a forceful and important role in natural language.

I have tried to give a reasonable sample of lexical ambiguity in my dictionary, but I certainly have not been as thorough as the most meager commercially available dictionary.

VIII. WORD CLASSIFICATIONS

Each word in the ERICA vocabulary has a grammatical classification string associated with it, according to the conventions described in VII above. Appendix 4 gives the dictionary for the complete corpus.

The same symbols are used for ERICA and the ADULT dictionaries. This is not to say that all the speakers necessarily have the same grammar or use language in the same way. The point is that they communicate, and our best hope of understanding how is to assume a common lexicon.

I include here both the fundamental syntactic categories and the entries that indicate multiple classification. Table 7 gives the categories and their intuitive meanings. Table 8 gives the entries as I have them in the dictionary. Table 9 breaks down the multiple classifications into the fundamental categories, counting for example words that could be used as nouns. Hence the numbers on Table 9 do not sum up to the total number of types in ERICA, which is 3,168.

TABLE 7

FUNDAMENTAL SYMBOLS USED IN THE DICTIONARY FOR ERICA AND THEIR INTUITIVE MEANINGS(*)

SYMBOL	MEANING AND EXPLANATION	EXAMPLE(S)
adj	common adjectives	good
adv	adverbs	well softly
aff	affirmative words	yes uhuh
art	articles	a an the
aux	auxiliary verbs	have did be
conj	conjunctions	and but
int	interjections	bye darn
intadv	interrogative adverbs	how when
inter	interrogative pronouns	who whom
link	linking verbs	be (and its inflections)
misc	miscellaneous words that defy classifiction (examining the contexts was unilluminating)	diller shafto
mod	modal verbs	can cause wanna
n	common nouns	house cat
neg	negating words	no not
padj	possessive adjectives made from either common or proper nouns	bear's erica's

---

* Recall that uppercase letters are mapped into lowercase.

persp	personal pronouns	i you him
pn	proper nouns	africa tom *
prep	preposition	except from
pron	pronouns other than personal and interrogative	anything someone
pronadj	adjectival form of a pronoun	his somebody's
qu	quantifying words and cardinal numbers	all both one two
v	verbs other than linking modal, and auxiliary	bake fit
<undef>	for unintelligible words and phrases	

TABLE 8

NUMBER OF WORD TYPES CLASSIFIED IN VARIOUS LEXICAL CATEGORIES

INCLUDING FUNDAMENTAL AND COMPLEX SYMBOLS

SYMBOL	FREQUENCY		
	CORPUS	ERICA	ADULT
n	1462	878	1337
v	651	354	601
adj	305	139	291
pn	161	96	143
adv	86	35	81
int	76	58	47
padj,n#aux,n#link	72	32	54
n,v	36	20	32
qu,pron	34	27	33
padj,pn#aux,pn#link	30	18	25
prep	23	16	22
misc	21	11	13
pron	19	13	15
mod	18	17	17
conj	16	8	15
persp	16	15	15
aff	15	12	10
pronadj	13	10	12
prep,adv	10	9	10
link,aux	8	7	8
persp#mod	8	5	8
mod#neg	7	5	7
persp#aux,persp#link	7	7	7
v,mod	7	6	6
pron#aux,pron#link	6	5	5
aux#neg,link#neg	5	4	4
neg	5	5	5
v,aux	5	5	5
intadv	4	4	4
v#neg,mod#neg	4	3	4
art	3	3	3
inter#aux,inter#link	3	3	2
n,adj	3	1	3
persp#v,persp#aux	3	0	3
qu	3	3	3
inter	2	2	2
mod#persp	2	2	2
prep,conj	2	2	2
pron#mod	2	1	1

<undef>	2	2	1
adv#link	1	1	1
intadv#mod	1	1	0
intadv#link	1	0	1
intadv#aux,intadv#link	1	1	1
inter#mod	1	1	0
inter#persp	1	1	0
inter,persp	1	0	1
mod#pron	1	1	0
n,adv	1	1	1
padj	1	1	1
persp,pronadj	1	1	1
pron#aux	1	0	1
v#persp	1	1	0

TABLE 9

FUNDAMENTAL SYMBOLS AND CONCATENATIONS IN THE ERICA DICTIONARY

SYMBOL	FREQUENCY		
	CORPUS	ERICA	ADULT
n	1502	900	1373
v	699	385	644
adj	308	140	294
pn	161	96	143
padj	103	51	80
adv	97	45	92
int	76	58	47
n#aux	72	32	54
n#link	72	32	54
pron	53	40	48
qu	37	30	36
prep	35	27	34
pn#aux	30	18	25
pn#link	30	18	25
mod	25	23	23
misc	21	11	13
conj	18	10	18
persp	18	16	17
aff	15	12	10
pronadj	14	11	13
aux	13	12	13
mod#neg	11	8	11
persp#aux	10	7	10
link	8	7	8
persp#mod	8	5	6
persp#link	7	7	7
pron#aux	7	5	6
pron#link	6	5	5
aux#neg	5	4	4
link#neg	5	4	4
neg	5	5	5
intadv	4	4	4
v#neg	4	3	4
art	3	3	3
inter#aux	3	3	2
inter#link	3	3	2
inter	3	2	3
persp#v	3	0	3
intadv#link	2	1	2
mod#persp	2	2	2

pron#mod	2	1	1
<undef>	2	2	1
adv#link	1	1	1
intadv#mod	1	1	0
intadv#aux	1	1	1
inter#mod	1	1	0
inter#persp	1	1	0
mod#pron	1	1	0
v#persp	1	1	0
Totals*	3,509	2,055	3,153

---

* The counts in this table represent the number of words that could take a certain grammatical class (fundamental or concatenation). Hence, the sums are greater than the actual number of words in the appropriate portion of the corpus.

IX. GOODNESS-OF-FIT TESTS ON THE ERICA AND ADULT DICTIONARIES

It is a reasonable hypothesis that the adult and child have similar frequencies of usage of words. Using the common 1,552 words of the ERICA and ADULT vocabularies, I constructed a 2-by-1,552 contingency table, and found that this hypothesis was untenable. With 1,551 degrees of freedom, the chi-square was 13,169.0460, which must be rejected at any reasonable level of significance.

While Erica and the adults do not use individual words with similar relative frequencies, they use words from the various grammatical categories in similar proportions. Thus, while the words 'dog' and 'cat' may, for example, be used more often by Erica than by the adults, nouns (any nouns) are used similarly. Table 10 gives that contingency table, showing a chi-square of 53.7626 for 53 degrees of freedom, roughly significant to 50 percent, obtained by taking the observed frequencies from the complete corpus as a predictor of the frequency in the ERICA portion alone. Table 11 shows the same results for predicting the ADULT frequencies from the complete corpus. This includes the grammatical classes that had fewer than 5 members, a practice that is usually bad form.

TABLE 10

PREDICTING ERICA LEXICAL CLASSES FROM ADULT LEXICAL CLASSES

LEXICAL CATEGORY	Adult Observed	Erica Observed	Erica Expected	Chi-square
n	1337	878	864.13	.22
v	601	354	388.44	3.05
adj	291	139	188.08	12.81
pn	143	96	92.42	.14
adv	81	35	52.35	5.75
int	47	58	30.38	25.12
padj,n#aux,n#link	54	32	34.90	.24
n,v	32	20	20.68	.02
qu,pron	33	27	21.33	1.51
padj,pn#aux,pn#link	25	18	16.16	.21
prep	22	16	14.22	.22
misc	13	11	8.40	.80
pron	15	13	9.69	1.13
mod	17	17	10.99	3.29
conj	16	8	10.34	.53
persp	15	15	9.69	2.90
aff	10	12	6.46	4.74
pronadj	12	10	7.76	.65
prep,adv	10	9	6.46	1.00
link,aux	8	7	5.17	.65
persp#mod	8	5	5.17	.01
mod#neg	7	5	4.52	.05
persp#aux,persp#link	7	7	4.52	1.35
v,mod	6	6	3.88	1.16
pron#aux,pron#link	5	5	3.23	.97
aux#neg,link#neg	4	4	2.59	.77
neg	5	5	3.23	.97
v,aux	5	5	3.23	.97
intadv	4	4	2.59	.77
v#neg,mod#neg	4	3	2.59	.07
art	3	3	1.94	.58
inter#aux,inter#link	2	3	1.29	2.26
n,adj	3	1	1.94	.45
persp#v,persp#aux	3	0	1.94	1.94
qu	3	3	1.94	.58
inter	2	2	1.29	.39
mod#persp	2	2	1.29	.39
prep,conj	2	2	1.29	.39

pron#mod	1	1	.65	.19
<undef>	1	2	.65	2.84
adv#link	1	1	.65	.19
intadv#mod	0	1	0.00	1.00
intadv#link	1	0	.65	.65
intadv#aux,intadv#link	1	1	.65	.19
inter#mod	0	1	0.00	1.00
inter#persp	0	1	0.00	1.00
inter,persp	1	0	.65	.65
mod#pron	0	1	0.00	1.00
n,adv	1	1	.65	.19
padj	1	1	.65	.19
persp,pronadj	1	1	.65	.19
pron#aux	1	0	.65	.65
v#persp	0	1	0.00	1.00

observed sum 1 = 2,867
observed sum 2 = 1,853
expected sum = 1,853.00
chi-square sum = 89.98

TABLE 11

PREDICTING ADULT LEXICAL CLASSES FROM ERICA LEXICAL CLASSES

LEXICAL CATEGORY	Adult Observed	Erica Observed	Erica Expected	Chi-square
n	878	1337	1336.44	.00
v	354	601	538.84	7.17
adj	139	291	211.58	29.81
pn	96	143	146.13	.07
adv	35	81	53.27	14.43
int	58	47	88.28	19.31
padj,n#aux,n#link	32	54	48.71	.57
n,v	20	32	30.44	.08
qu,pron	27	33	41.10	1.60
padj,pn#aux,pn#link	18	25	27.40	.21
prep	16	22	24.35	.23
misc	11	13	16.74	.84
pron	13	15	19.79	1.16
mod	17	17	25.88	3.04
conj	8	16	12.18	1.20
persp	15	15	22.83	2.69
aff	12	10	18.27	3.74
pronadj	10	12	15.22	.68
prep,adv	9	10	13.70	1.00
link,aux	7	8	10.65	.66
persp#mod	5	8	7.61	.02
mod#neg	5	7	7.61	.05
persp#aux,persp#link	7	7	10.65	1.25
v,mod	6	6	9.13	1.07
pron#aux,pron#link	5	5	7.61	.90
aux#neg,link#neg	4	4	6.09	.72
neg	5	5	7.61	.90
v,aux	5	5	7.61	.90
intadv	4	4	6.09	.72
v#neg,mod#neg	3	4	4.57	.07
art	3	3	4.57	.54
inter#aux,inter#link	3	2	4.57	1.44
n,adj	1	3	1.52	1.43
persp#v,persp#aux	0	3	0.00	9.00
qu	3	3	4.57	.54
inter	2	2	3.04	.36
mod#persp	2	2	3.04	.36

```
prep,conj                   2         2         3.04       .36
pron#mod                    1         1         1.52       .18
<undef>                     2         1         3.04      1.37
adv#link                    1         1         1.52       .18
intadv#mod                  1         0         1.52      1.52
intadv#link                 0         1         0.00      1.00
intadv#aux,intadv#link      1         1         1.52       .18
inter#mod                   1         0         1.52      1.52
inter#persp                 1         0         1.52      1.52
inter,persp                 0         1         0.00      1.00
mod#pron                    1         0         1.52      1.52
n,adv                       1         1         1.52       .18
padj                        1         1         1.52       .18
persp,pronadj               1         1         1.52       .18
pron#aux                    0         1         0.00      1.00
v#persp                     1         0         1.52      1.52


observed sum 1 = 1,942
observed sum 2 = 2,956
expected sum =   2,909.53
chi-square sum =   212.14
```

CHAPTER 3 — FORMAL DEVELOPMENTS

I. GENERATIVE GRAMMARS

This chapter is devoted to standard concepts and results of the theory of generative grammars as well as some notational matters.

Let V be a set of symbols. Then, V^* is the set of all finite sequences of elements of V, including the empty string, which is denoted by ε. Such finite sequences are sometimes called strings.

$V+$ denotes $V^* - \{\varepsilon\}$. Small letters a,b,c are variables ranging over members of V^*.

A structure

$$G = \langle V,T,S,P\rangle$$

is a generative grammar just in case G satisfies the conditions:

1) V is a finite nonempty set of symbols, the vocabulary;

2) T is a nonempty subset of V, known as the terminal vocabulary;

Then, let the nonterminal vocabulary $V_N = V-T$.

3) S is a distinguished element of VN, called the start symbol;

4) P, the set of productions or rules, is a finite subset of the set V+ X V*.

Let T+ be the set of all finite non-empty terminal strings. Further, if <a,b> $\in$ P, then I write (informally)

a -> b

to indicate that this is a production in P. The symbol a is the left-hand side (lhs) of <a,b> and b is the right-hand side (rhs) of <a,b>.

If a,b are strings in V*, then b is immediately produced from a if and only if there is a subsequence a′ in a and a subsequence b′ in b such that b is the result of substituting b′ in a for a′, and such that

a′ -> b′

is a rule in P. The intuition here is that an immediate production is what one obtains by replacing into some string for the left-hand side of some production by the right-hand side of that production.

If a,b are in V^* , then b is <u>derivable from a</u> if and only if there exist

a1, a2, ..., an for some n

such that

a (immediately) produces a1

a1 produces a2

a2 produces a3

.

.

.

.

an produces b.

The sequence <a,a1> <a1,a2>, ... ,<an,b> is called a <u>derivation of b from a</u>.

As an example of these ideas, consider the following grammar G that generates a few English sentences.

G = <V,T,S,P>

where

V = {S,NP,VP,N,ART,V,a,the,boy,girl,sees,knows,runs}

and

T = {a,the,boy,girl,sees,knows,runs} ;

hence, the set VN of non-terminals is

VN = {S,NP,VP,N,ART,V};

S is the start symbol (for <u>sentence</u>)

and P contains the rules

S -> NP VP
NP -> N
NP -> ART N
VP -> V
VP -> V NP
N -> boy   N -> girl
ART -> a   ART -> the
V -> runs   V -> sees   V -> knows

Hence, S produces NP VP. Also, the string

the boy

is derivable from the string NP. This relationship is denoted by

$$\mathrm{NP} \underset{G}{\Longrightarrow} \text{the boy}$$

where G (a reference to the grammar) may be omitted when the grammar is clear.

The set of noun phrases is the set of all terminal strings derivable from the symbol NP. What we are interested in is the set of terminal strings in T+ that is derivable from the start symbol, i.e.,

$$\{a \in T+ \mid S \underset{G}{\Longrightarrow} a\}$$

This is the *language of the grammar G*, denoted by L(G). Usually, when I say 'derivation' I mean derivation from the start symbol to a terminal string. If grammars G1 and G2 are such that L(G1)=L(G2), then G1 is said to be equivalent to G2.

The following strings are in L(G):

boy runs
the boy runs
the boy sees the girl
the girl sees the boy

Notice that the definition of derivation allows several sequences that are derivations for 'boy runs'. Two of them are:

1) <S,NP VP> <NP VP,N VP> <N VP,N V> <N V,boy V>
<boy V,boy runs>

2) <S,NP VP> <NP VP,NP V> <NP V,NP runs> <NP runs,N runs>
<N runs,boy runs>

In the above, 1) and 2) differ only in the "order" that the rules are applied, and they seem to be "one derivation in two different orders". What is needed is a notion of "derivation" that selects only one of these. The notion I use is that of a left-most derivation.

A derivation is a left-most derivation just in case, in each pair of the sequence, the substitution is made for the left-most possible sequence of symbols from which a substitution could be made. Notice that 1) is left-most, and 2) is right-most, admitting the symmetric concept. The concept of left-most derivation is not readily useful with all kinds of grammars.

Different kinds of generative grammars are obtained by putting restrictions on the production rules that may be in P. A type-0 or *recursively enumerable* grammar has no further restrictions placed upon it. A *type-1* or *context-sensitive* grammar has only the restriction that if <a,b> is in P then $|b| >= |a|$, where $|a|$ is the number of symbols in a, the *length of a*. A *type-2* or *context-free* grammar is context-sensitive plus if <a,b> is in P then $|a|=1$; further, only non-terminals may occur on the left-hand side of the derivation. (In fact, it is sometimes the practice to define the classes of terminals and non-terminals from the productions in a context-free grammar. This is the way a compiler would handle the compilation of a program in, say, ALGOL.) Notice that the above grammar G is context-free. A *type-3* or *regular* grammar is context-free, plus if <a,b> is in P then b is either of the form

t

or of the form

tN

where t is a terminal and N is a non-terminal. In addition, other grammars of various intermediate strengths are possible.

I am concerned exclusively with context-free grammars. These grammars are easily created and parsing programs can be easily written for context-free grammars. (Usually I say 'cfg' for 'context-free grammar', 'cfl' for 'context-free language'.) Moreover, set-theoretical semantics applies very naturally to cfg.

For cfg, it can be shown that if a string has any derivation, then it has a left-most derivation. The sense of "one derivation in several different orders" is correctly captured by the notion of left-most derivation. When I say 'derivation', unless otherwise noted, I mean 'left-most derivation'.

II. THE RELATION OF GENERATIVE GRAMMARS TO AUTOMATA

A conceptually important fact is that the relation between the theory of generative grammars and the theory of automata is well understood (1). I shall say that an automaton recognizes a language if and only if the automaton, given an input string, stops and returns a TRUE if the string is in the language. In particular, regular languages are representable by finite automata (and conversely); and context-free languages are representable by push-down automata (and conversely). Every context-sensitive language is recognized by some Turing machine that always halts, so that context-sensitive languages are recursive. The converse is not the case, however, since there are recursive sets that are not context-sensitive languages. Each type-0 language is recognized by some Turing machine, but the machine may not necessarily halt on a string not in the set in question (hence the name "recursively enumerable").

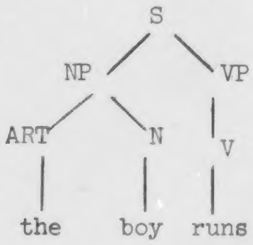
---

(1) See, for example, Hopcroft and Ullman, Formal Languages and Their Relation to Automata, Reading, Mass., 1969.

III. DERIVATIONS AND TREES

While the notion of a left-most derivation is the formal definition of "derivation" that I want to use, informally the concept of a _tree_ (2) is far superior. I take it that the idea of a tree is sufficiently intuitive to require no further explanation, except to give a few examples.

In the above example of the cfg G, consider the derivation of 'boy runs'. This can be represented by the tree



---

(2) See [Suppes-2] for a tree-oriented approach.

Note that each of the (non-left-most) derivations yields this same tree. It is possible to define the notion of tree and proceed to show that, for cfg, there is a one-one correspondence between left-most-derivations and trees.

It may happen that there are two or more left-most derivations for a string according to a cfg. Consider the grammar G' obtained from G above by adding the rule

S -> ART N VP

Then, the sentence

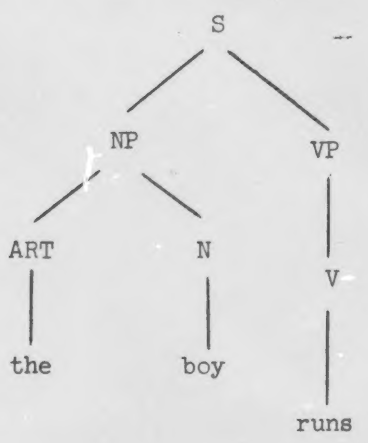
the boy runs

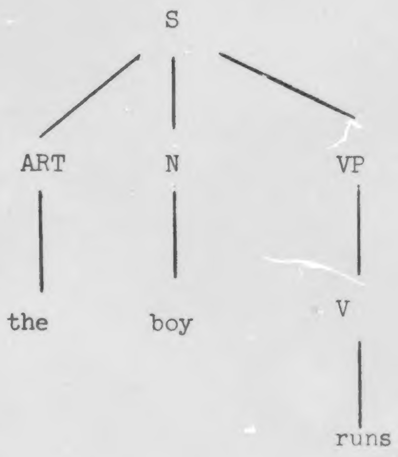
has two leftmost derivations:

1) <S,NP VP> <NP VP,ART N VP> <ART N VP,the N VP>
<the N VP,the boy VP>
<the boy VP,the boy V> <the boy V,the boy runs>

2) <S,ART N VP> <ART N VP,the N VP> <the N VP,the boy VP>
<the boy VP,the boy V> <the boy V,the boy runs>

Each derivation is represented by a different tree, viz.:





When a string has two or more (left-most) derivations, the string is said to be *grammatically ambiguous*. A grammar G is *grammatically ambiguous* if and only if some string in L(G) is grammatically ambiguous.

As a notational device, partition the set P of productions into *rule classes* such that all elements of the same rule class have the same lhs. Then number (arbitrarily) the classes in the partition so that each lhs has a number i, and further, give each rule in each class a number j. Thus, a rule is uniquely represented by the pair (i,j), called the *label* of the rule; and all rules having the same lhs have the same number i, and no two rules with i as the first element of the label have the same number j as the second element of the label. It is then possible to denote a derivation by a sequence of labels (assuming that we are starting with the start symbol and that the derivation will be leftmost.)

If I label the rules in G by this scheme:

(1,1)	s -> np vp
(2,1)	np -> n
(2,2)	np -> art n
(3,1)	vp -> v
(3,2)	vp -> v np
(4,1)	art -> a
(4,2)	art -> the
(5,1)	n -> boy
(5,2)	n -> girl
(6,1)	v -> runs
(6,2)	v -> sees
(6,3)	v -> knows

then the left-most derivation of 'the boy sees the girl' may be represented by the label sequence

(1,1) (2,2) (4,2) (5,1) (3,2) (6,2) (2,2) (4,2) (5,2) .

IV. CHOMSKY NORMAL FORM GRAMMARS

If a cfg G is such that each rule in P is either of the form

A -> a

or of the form

A -> B C

then G is said to be in Chomsky normal form. Every cfg has an equivalent grammar that is in Chomsky normal form. Moreover, it is possible, given a Chomsky normal form grammar G' that represents a grammar G, to obtain a derivation in G from a derivation in G'.

V. LEXICAL SIMPLIFICATION OF CONTEXT-FREE GRAMMARS

My syntactic theory for the ERICA corpus is highly dependent on the use of a dictionary to classify words according to grammatical categories. When an utterance is to be parsed by the grammars I use, the utterance is first converted to its lexical form (which may be 'shorthand' for several alternative forms). The grammar then sees only the alternative forms and never sees the original utterance. The vocabulary V of the grammar does not contain the actual words in the utterance but only symbols for the grammatical categories, plus additional symbols.

It represents a philosophical-psychological question as to whether the dictionary exists separately from the grammar (as I believe) or as only a shorthand for rules in the grammar. I will discuss this further in Chapter 4.

I shall say that G _admits of lexical simplification_ just in case:

1) there is a non-empty subset DP of the set of rules P such that for each $p \in DP$, p is of the form

$$A \rightarrow d$$

where A is a non-terminal, and d is a terminal of G;

2) let $D = \{ d \mid A \rightarrow d \text{ is in } DP \}$, called the set of _lexical symbols_. Then, no $d \in D$ occurs in any rule in $P - DP$.

Many of the grammars useful for natural language admit to lexical simplification.

The gain, computationally, is that a different procedure can be used on the set D of symbols than the procedure used for the grammar as a whole, provided a large number of symbols get put into the class D (3).

Clearly cfg exist that cannot be lexically simplified. One such case is the grammar consisting of the following productions:

(1,1) S -> A B
(1,2) S -> a c
(1,3) S -> b c
(1,4) S -> a
(1,5) S -> b

No non-empty set DP can be constructed, since the symbols a and b occur in rules (1,2) and (1,3) respectively. Hence, adding a lexicon to this grammar is impossible. A different grammar for the same language would, perhaps, allow a lexicon. But the lexicon should not change the structure of derivations in the language, only simpify them.

The conceptually interesting fact about lexical simplification is that it can greatly reduce the 'parsing' machinery when the surface language has a very large vocabulary that can be classified (perhaps with great overlapping) into a relatively small number of "grammatical categories". Moreover, if this is happening we have, among other things, the basis for probabilistic theories of sentence production, based upon the probability of uttering lexical forms rather than actual strings of words.

---

(3) Programming languages such as ALGOL-60 often have their syntax defined in terms of context-free grammars. According to such definitions, one would believe that the parser for an ALGOL-60 compiler ran straight through the derivation of the "program" during compilation. In fact, this is not the case with any actual compiler I am familiar with. In practice, compilers take advantage of many things about the language in order to gain greater efficiency. An example is the search for numbers and arithmetic expressions in the program. This search is customarily implemented by a different routine that looks especially for expressions, and replaces them before the actual parser sees them. This is analogous to having a dictionary system for natural language.

CHAPTER 4 — A GRAMMAR FOR ERICA

I. THE SIMPLE MODEL

There is a straightforward way to generate a probability space from a cfg: assign a non-zero parameter to each rule in the grammar and require that the parameters for each class of rules with a given left-hand-side sum to 1. It is easy to see that this generates a non-zero probability for each sentence in L(G), and that the sum of the probabilities over L(G) (possibly an infinite set) is 1.

For example, consider the grammar G

G = <V,T,NP,P> , where

V = { NP,ADJ,ADP,N }

and

T = { ADJ,N }

and P has the rules

(1,1) NP -> N
(1,2) NP -> ADP N
(2,1) ADP -> ADJ
(2,2) ADP -> ADP ADJ

(this is a noun-phrase grammar).

Then L(G) is infinite since rule (2,2) may be applied recursively so that for each natural number n,

$$\mathrm{ADP} \underset{G}{\Longrightarrow} \underset{n\ \text{times}}{\mathrm{ADJ}\ \dots\ \mathrm{ADJ}}$$

(sometimes denoted ADJ ^n)

and hence

$$\mathrm{NP} \underset{G}{\Longrightarrow} \mathrm{ADJ}\ \hat{}n\ \mathrm{N}$$

for each natural number n.

Suppose we assign the following probabilities to the rules in P:

DISTRIBUTION D TO RULES IN GRAMMAR G

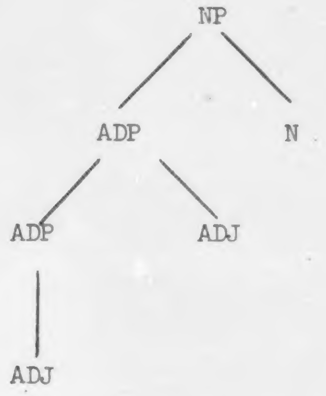
Rule	Probability
(1,1)	.6
(1,2)	.4
(2,1)	.7
(2,2)	.3

(this is not unreasonable);

then the noun-phrase

*) ADJ ADJ N

is parsed by the tree T*:



I shall say that the conditional probability of applying rule (i,j) given that some rule in the i-class is to be applied is the parameter associated with (i,j), and I denote this parameter b[i,j]. The probability associated with a tree T is the product of the parameters of the sequence of rules that generates T. Hence, the probability of T* is the expression:

P*) b[1,2]*b[2,2]*b[2,1]

which evaluates to .084 for the distribution D given above.

II. PROBABILITY AND LINGUISTICS

While L(G) is infinite, the probability of generating the noun-phrases of increasing length decreases geometrically. Most of the probability is represented by the noun-phrases in the following list:

NOUN PHRASE	PROBABILITY (by Distribution D)
n	.6
adj n	.28
adj adj n	.084
adj adj adj n	.0252
total	.9892

Thus only about one percent is shared by the remaining infinitely many noun phrases in G under the distribution. It is the thinness of the tail of the distribution of noun phrases (or sentences) that makes it plausible to use cfg in predicting finite samples of speech. The importance of this point is that it commits us to dealing probabilistically if we are to make sense of the idea that cfg can describe linguistic behaviour. Noam Chomsky (1) often proposes infinite grammars as models for speech (though he might not say it was a model), but at the same time shuns probabilistic treatments of grammar as being inappropriate. The data, however, are clear on this much: given a system (such as my dictionary) for classifying sentences, the noun-phrase

ADJ N

is more likely than

ADJ ADJ N

and

ADJ ^1000 N

has virtually no likelihood of being found. So we clearly cannot hold that all sentences in L(G) are equally likely. If we want to examine the phenomenon at all, the only plausible explanation, given the acceptability of context-free grammars as models for speech, is to affix a probability measure to the rules of the grammars used to model the speech.

---

(1) See Noam Chomsky, "Quine's Empirical Assumptions" in *Words and Objections: Essays on the Work of W. V. Quine*, D. Davidson and J. Hintikka (editors), Dordrecht, Holland, 1969, pp. 53-68.

In the event that a given sentence-type has two or more trees generated by a grammar G , then G is said to be grammatically ambiguous (cf. Chapter 3). A probability distribution on a grammar generates a probability for each tree. When a sentence-type has two or more trees, the obvious solution is to sum together the probabilities of the trees.

For example, if we add rule (2,3)

(2,3) ADP -> ADJ ADJ

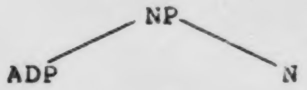
to G above, then the probability of *) is given by

*)'' b[1,2]*b[2,2] + b[1,2]*b[2,3]

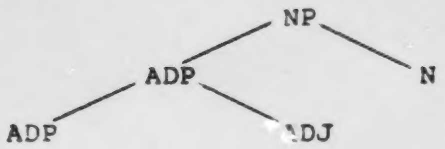
where b[2,3] is the probability of (2,3). (Of course, Distribution D cannot be used unless b[2,3] is 0. If a rule (i,j) is to have probability 0, it is a superfluous rule in the present context.)

The question may quite appropriately arise: why the particular probabilistic model imposed by fixing a probability on each rule? The answer, I believe, is inherent in the idea that the notion of cfg tries to capture, if not in the formal definition itself. The idea of a cfg is that a given rule (i,j) is used to replace its left-hand side without regard to the rest of the tree into which the replacement is made. Consider the (partial) trees T1 and T2 in the grammar G:

T1:



T2:



in relation to the rule

(2,1) ADP -> ADJ .

If we suppose that (2,1) has a probability p1 of being applied to T1 , and p2 of being applied to T2 , such that p1 is not equal to p2, then I would claim that the underlying grammar is actually context-sensitive since we are apparently looking at the "context" to determine which probability is appropriate. A proof of the claim might be to show that an algorithm suitable for determining which probability to use would not be calculable, in general, by a push-down automaton. Whether this would be completely persuasive or merely begging the question is debatable.

Talking about the probability of generating a particular sentence inherently uses "performance" language and standards, in that we are providing a model for observed linguistic behaviour. As much as one might be disposed to finding this an inappropriate approach to the philosophy of language, there is this much to either account for or dismiss: it is commonplace to assert that some things are more likely to be said than others, and the hard evidence supports this completely.

This point can be illustrated by looking at two recursive rules from the grammar GE1 that I have developed for use with ERICA (see Table 3 for the complete GE1). The rules are:

(1,2) ADJP -> ADJP ADJ
(14,2) ADVP -> ADV ADVP

Rule (1,2) is the recursive adjective phrase rule, and rule (14,2) is the recursive adverbial phrase rule. Tables 1 and 2 give the sentences in the ERICA corpus that required these rules (2).

---

(2) The method used for obtaining these results will be explained later in this chapter. The point of introducing the results ahead of their explanation is to make a point in regard to the low probability of long strings of adjectives and adverbs. Incidently, it is implausible, looking at the results in Chapter 2 on the length of utterances, that the length of the utterance alone is a good predictor of the number of, say, adjectives used. The fact is that the tendency to use repeated adjectives drops off more quickly than the tendency to increase length would indicate.

TABLE 1

SENTENCES IN ERICA THAT REQUIRE THE RECURSIVE ADJECTIVE PHRASE RULE

GRAMMAR GE1

RULE: (1,2) adjp -> adjp adj

FREQ	No. of TREES	Sentence Type	No.of Usages of Rule (1 if blank)
11	1	adj adj n	
7	1	adj adj adj	2
6	1	adj adj	
2	1	neg adj adj	
2	1	persp link art adj adj n	
2	1	pron link art adj adj n	
1	1	adj adj n n	
1	1	adj adj pron	
1	1	adj adj adj n	2
1	1	adj adj n v prep	
1	1	adj adj n v art n	
1	1	adj adj adj adj adj	4
1	1	adj adj n prep art n	
1	1	adj adj n mod neg v qu n	
1	1	adj adj adj adj adj adj adj adj	7
1	1	adj adj n conj pron aux v art n	
1	5	adv adv adj adj	5 (one per tree)
1	1	adv link art adj adj pron	
1	1	art adj adj	
1	1	art adj adj n	
1	1	art adj adj n v	
1	1	art adj adj pron	
1	1	art adj adj adj n	2
1	1	conj art adj adj n	
1	1	conj pron link adj adj pron	
1	1	conj persp v art adj adj pron	
1	1	conj pron link art adj adj pron	
1	1	int pron link art adj adj n	
1	1	n link art adj adj n	
1	1	persp link adj adj	
1	1	persp art adj adj n	
1	1	persp v art adj adj n	
1	1	persp link neg adj adj adj	2
1	1	persp link neg qu adj adj adj	2

```
1        1       pn link art adj adj n
1        1       pron link adj adj
1        1       pron art adj adj n
1        1       pron link art adj adj adj                2
1        1       qu adj adj n

SENTENCE TYPES = 39    TOKENS = 63

TIMES RULE (1,2) WAS USED  = 58
TIMES USED*FREQUENCY OF SENTENCE = 88

NOTE:   Due to grammatical ambiguity in the corpus,
the above statistics may be misleading.
```

TABLE 2

SENTENCES IN ERICA THAT REQUIRE THE RECURSIVE ADVERBIAL PHRASE RULE

GRAMMAR GE1

RULE: (14,2) advp -> adv advp

FREQ	No. of Trees	SENTENCE TYPE	No. of Usages of Rule (14,2)
13	1	adv adv	
9	1	inter link adv adv	
2	2	persp link adv adv adj	
1	1	adv adv adv	2
1	2	adv adv adj n	
1	5	adv adv adj adj	2
1	2	persp link adv adv adj pron n	
1	2	pron link adv adv adj	

SENTENCE TYPES = 8 SENTENCE TOKENS = 29
TIMES RULE (14,2) WAS USED = 10
TIMES USED*FREQUENCY = 31

TABLE 3

GRAMMAR GE1

LABEL	RULE
(1,1)	adjp -> adj
(1,2)	adjp -> adjp adj
(1,3)	adjp -> advp adjp
(14,1)	advp -> adv
(14,2)	advp -> adv advp
(21,1)	quart -> qu
(21,2)	quart -> art
(9,1)	adp -> adjp
(9,2)	adp -> det
(9,3)	adp -> det adjp
(22,1)	qadp -> adjp
(22,2)	qadp -> quart adjp
(22,3)	qadp -> quart
(22,4)	qadp -> det
(22,5)	qadp -> det adjp
(10,1)	det -> pronadj
(10,2)	det -> padj
(2,1)	nounp -> pn
(2,2)	nounp -> n
(2,3)	nounp -> pron
(13,1)	np -> npsub prepp
(13,3)	np -> npsub conj npsub
(13,4)	np -> npsub
(17,1)	npsub -> persp
(17,2)	npsub -> nounp
(17,3)	npsub -> adp nounp
(17,4)	npsub -> quart nounp
(17,5)	npsub -> quart adjp nounp
(5,1)	vbl -> auxilp vp
(5,2)	vbl -> vp
(16,1)	auxilp -> auxil
(16,2)	auxilp -> auxil neg
(15,1)	auxil -> aux
(15,2)	auxil -> mod
(3,1)	vp -> verb
(3,2)	vp -> verb prep
(3,3)	vp -> verb np
(3,4)	vp -> verb np np

(3,5)	vp -> verb prepp np
(3,6)	vp -> verb np prepp
(3,8)	vp -> verb np prep
(3,9)	vp -> verb prepp
(11,1)	verb -> v
(11,2)	verb -> v neg
(19,1)	linkp -> link
(19,2)	linkp -> link neg
(7,1)	nom -> npsub prepp
(7,3)	nom -> npsub conj npsub
(7,4)	nom -> nom1
(7,5)	nom -> qadp
(18,1)	nom1 -> npsub
(18,2)	nom1 -> nom1 npsub
(4,1)	a -> nom
(4,2)	a -> inter
(4,3)	a -> subj vbl
(4,4)	a -> inter vbl
(4,5)	a -> subj linkp prepp
(4,6)	a -> inter linkp
(4,7)	a -> mod subj
(4,8)	a -> prepp
(4,9)	a -> linkp subj qadp
(4,10)	a -> linkp subj np
(4,11)	a -> subj linkp np
(4,12)	a -> subj linkp qadp
(4,13)	a -> auxilp subj vp
(4,14)	a -> subj np vbl
(4,15)	a -> subj linkp np np
(4,16)	a -> auxilp subj np
(4,19)	a -> auxilp subj
(4,20)	a -> verb
(4,21)	a -> intadv auxilp subj vp
(4,22)	a -> intadv auxilp subj
(4,23)	a -> intadv
(4,24)	a -> verb subj
(4,25)	a -> advp subj auxilp
(4,28)	a -> subj auxilp
(4,29)	a -> advp
(4,30)	a -> inter subj
(4,31)	a -> inter linkp subj
(4,32)	a -> inter np vbl
(4,33)	a -> advp subj vbl
(4,35)	a -> vbl subj prep
(4,37)	a -> verb subj np
(4,38)	a -> intadv subj vbl
(4,39)	a -> auxilp v
(4,40)	a -> advp linkp subj
(4,41)	a -> linkp qadp
(4,42)	a -> inter linkp advp

(4,43)	a -> subj vp auxilp
(4,44)	a -> inter auxilp np verb
(4,45)	a -> subj linkp
(4,46)	a -> inter auxilp advp
(12,1)	prepp -> prep np
(6,1)	subj -> np
(6,2)	subj -> np prepp
(8,1)	s -> a
(8,2)	s -> aff int
(8,3)	s -> int aff
(8,4)	s -> neg a
(8,5)	s -> aff a
(8,6)	s -> a aff
(8,7)	s -> neg
(8,8)	s -> aff
(8,9)	s -> int
(8,10)	s -> conj
(8,11)	s -> aff aff
(8,12)	s -> int int
(8,15)	s -> neg neg
(8,16)	s -> conj a
(8,17)	s -> a conj
(8,18)	s -> int a
(8,19)	s -> a int

Tables 1 and 2 show the following trend in the sentences that use the recursive adjective/adverb phrase rules: the tendency to use the rules repeatedly is small. Table 4 shows the type/token counts for the repeated usages of these rules.

TABLE 4

REPEATED USAGES OF RECURSIVE RULES (1,2) AND (14,2)

RULE (1,2)

NO. OF TIMES USED	TYPES	TOKENS
1	31	49
2	6	12
3	0	0
4	1	1
5	0	0
6	0	0
7	1	1
Totals	39	63

NOTE: This counts sentence type
adv adv adj adj
only once, rather than counting for each of
the 5 ambiguities.

RULE (14,2)

1	7	28
2	1	1
	8	29

NOTE: This count uses for each sentence type the grammatical ambiguity that had the most usages of rule (14,2).

III. MAXIMUM LIKELIHOOD AND ESTIMATIONS

If S is a set of sentence types, together with a non-zero frequency for each sentence type, then S is a sentence sample. The question, "How well does cfg G describe the syntax of sample S?" is one that can be given meaning in terms of a probability distribution on G. Several kinds of tests are available to determine the "goodness of fit" of G to S. Among these, the method of maximum likelihood stands out for its well-understood properties. The method involves two steps: 1) estimating the parameters (in this case, the b[i,j] 's) so that the probability of S given G is a maximum; and 2) using some test to evaluate the discrepancy between the observed frequencies in the sentence sample S and the expected or theoretical frequencies provided by the estimated parameters.

Given any assignment of probabilities $b[i,j]$ to the rules of G, such that for all i,

$$\sum_j b[i,j] = 1$$

we have a probability for any sample S. If k is in S, let FREQ(k) be the frequency associated with k. Assume that no k in S is a lexically ambiguous form (see Chapter 3). Then let PROB(k) be the expected probability of k, computed by first finding the probability of each tree for k and then summing over the probabilities for all such trees, as above. The probability of S is then given by the likelihood equation:

$$L) \quad \prod_{k \in S} PROB(k)^{FREQ(k)} \quad .$$

If G is grammatically unambiguous, then for each k in S, PROB(k) is a product of some of the $b[i,j]$'s, and the problem of finding values for the $b[i,j]$'s that maximize L has a simple analytical solution (3).

If there are j rules in class i, then we shall say that this class contributes j-1 independent parameters. (This is because the rules must sum to 1 in each class.)

---

(3) See [Suppes-1] for a simple derivation. The solution is obtained by taking the ln(L) (the natural logarithm), computing the partial derivatives with respect to the parameters, and solving the resulting set of equations.

For the analytic solution, we need a simple concept, the USAGE(i,j) of rule (i,j). For each i,j, let USAGE(i,j) be the number of times that rule (i,j) is used in deriving the sentences in S, weighted by the frequencies. For example, if the rule (i,j) is used on three sentences k1, k2, and k3, with frequencies f1, f2, and f3, and supposing that rule (i,j) is used twice on k3, then USAGE(i,j) is

$$f1 + f2 + 2*f3 \quad .$$

The analytical solution then gives us an estimate for each b[i,j], the parameter associated with rule (i,j), by the formula

$$b[i,j] = \frac{USAGE(i,j)}{\sum_{i} USAGE(i,j)}$$

The b[i,j] 's then are such that L is at a maximum (4).

Let G be *grammatically ambiguous relative to a sample S* if and only if for some k in S, k has two or more G-derivations. (Notice that the above maximum likelihood solution requires only non-ambiguity of the grammar relative to sample S under consideration.) If, however, G is relatively ambiguous, then the analytical solution to the maximum likelihood problem is not known, to the best of my knowledge. In general, the expressions for the probability of a given k in S will be the sum of products, and the terms of the maximum likelihood equation become quite complicated.

---

(4) The solution to the maximum likelihood problem for the unambiguous case generates only probabilities that are in the interval [0,1], which is, of course, the meaningful range for probabilities. Maximum likelihood methods often have to contend with solutions outside of this region.

In an effort to approximate the solutions to these equations I have used a numerical analysis program called MINFUN (5).

In my experience, a reasonable approximation appears to arise from what I call the *equal weights* approximation method. Consider a sentence-type k with n trees, and notice that if we had the appropriate weights for each of the n trees, we could use them to divide up the observed frequency of k and thus compute the correct USAGE(i,j) for each rule (i,j). If there is only a limited amount of grammatical ambiguity (say, less than 5 percent of S), then to weight equally the n trees for terminal-form k in S seems to give values for the $b[i,j]$'s that are very little different from MINFUN-generated values. (Originally, I used the equal weights method to prepare initial values for MINFUN, and found very little improvement even after hours of searching the probability space for improved values.)

IV. CHI-SQUARE AND GOODNESS OF FIT TESTS

Any parameter estimation fixes a probability on each sentence type k in S. It remains to test the goodness of the fit. I used two main methods augmented by several other statistical procedures. The main methods are the *chi-square* and *modified chi-square* tests.

---

(5) I would like to thank Mr. Clark Crane of the Stanford Computer Science Department for permission to use his program MINFUN for this purpose. MINFUN was written in OS/Fortran for the IBM 360/67. I rewrote it for use on the PDP-10 in Fortran IV.

MINFUN estimates the maximum likelihood values for the parameters by being fed the negative logarithm of the maximum likelihood equation, as well as the partial derivatives thereof. I wrote several programs to perform this monumental equation writing and symbolic differentiation, passing the equations to the FORTRAN compiler for linkage to MINFUN by the loader. Details of this process are available on request, but are not included here due to their basic irrelevance.

To resolve the equations that are generated by even a small sample S (say, the sentence types in ERICA with frequency >= 5) requires a great deal of computation by MINFUN. To deal with the entire distribution is quite impossible. Each new grammar requires completely new analysis.

With 75-plus independent variables, this problem is quite messy by the MINFUN program. I have experimented with several other programs, however, and only MINFUN has the necessary understanding of the problem of *forbidden regions*, which arises when parameters pass into values representing physically or conceptually impossible situations (here, the forbidden region is probabilities outside the region [0,1]).

The chi-square test is well-known for its distributional properties. Let SUM be the sum of the frequencies of all k in S, and let EXP(k), the *expected frequency of k*, be

$$\mathrm{SUM} * \mathrm{PROB}(k) \quad .$$

The *chi-square contribution of k* is given by the formula

$$\mathrm{CHISQUARE}(k) = \frac{(\mathrm{FREQ}(k) - \mathrm{EXP}(k))^2}{\mathrm{EXP}(k)} \quad .$$

I shall say (somewhat imprecisely) that the *chi-square statistic associated with a model* is the sum over k of CHISQUARE(k).

Tables of the level of significance of the chi-square test are commonly available in any statistics text.

To compute the level of significance, another important factor is the *degrees of freedom*. Intuitively, this is the number of things that are being predicted by the model. It is the number of sentence types less the number of independent parameters in the model, less 1 (since the fact that the probability must sum to 1 removes a degree of freedom). The number of independent parameters is

$\sum_{i}$ $(j-1)$ such that there are j rules with the label (i,k), for some k

Some of the problems associated with using the chi-square test are:

1) The test should not be applied to sentence-types k such that $EXP(k) < 5$. This is a rule of thumb resulting from the problem that S is a discrete distribution while the chi-square is based on a continuous distribution. To counteract this problem, my estimating program grouped together the expected and observed frequencies of sentence-types k where $EXP(k) < 5$. The grouping was done somewhat arbitrarily. I am not really happy with this solution of grouping unless the sentence-types can be grouped according to some criterion that makes the group plausible.

2) The chi-square test is unrealistically sensitive to sentence-types with smaller expected frequencies. This is because the chi-square is a continuous distribution, but the applications often made are to discrete distributions, as is the case here. An attempt often made to correct for this manifestation of the continuous nature of chi-square is to subtract a small value from the term

$$| FREQ(k) - EXP(k) |$$

used in the numerator of CHISQUARE(k). This correction for continuity has little effect on the cells at the top of the distribution; it is largely felt at the bottom where the disparity between the discrete and continuous distribution is greatest.

The second method used for determining the goodness of fit is the modified chi-square, which simply reverses the role of EXP(k) and FREQ(k). The contribution of k to MCHI2 is

$$\mathrm{MCHI2}(K) = \frac{(\mathrm{FREQ}(k) - \mathrm{EXP}(K))^2}{\mathrm{FREQ}(k)}$$

The point of the modified chi-square is to minimize the effect of a few cells with very small expected frequency.

V. GEOMETRIC MODELS FOR CFG

The model for a cfg that has j-1 independent parameters for each class i of rules of cardinality j is called the full parameter model. It is, however, possible to use only one parameter per class by ranking rules (i,j) according to USAGE(i,j) and applying one of several distributions that use only one parameter. In

Appendix 1, several models for the length of utterances in ERICA are discussed. Examination of the properties of the several distributions used in Appendix 1 (geometric, poisson, negative binomial) quickly reveals that the geometric is the most plausible. The method I used for applying the geometric distribution to cfg is: order the rules (i,j) in a given class i, remove unused rules (which therefore have probability 0), and apply the geometric distribution--i.e., with a single parameter b the probability assigned to the top rule in the class is (1-b); to the next, b*(1-b); to the third, b^2*(1-b), and so on. The last rule gets all the remaining probability, hence the distribution is a truncated geometric. Then solve for the value of b that maximizes the probability that the USAGE distribution was obtained, given the geometric model.

Most classes of rules lend themselves quite well to the geometric model, and the chi-squares are little different. The gain, statistically speaking, is in the number of independent parameters involved in the model. Some classes of rules have 40 members, and to predict the USAGE's of all these with only one parameter is somewhat impressive. Conceptually, it suggests a mechanism for syntax generation based on the class of rules that can effect a certain replacement (e.g., the rules that replace the noun phrase with a pronoun, a noun, a determiner-noun, etc.).

Since various models have different numbers of parameters, the best overall comparison I offer is the chi-square (or modified chi-square) divided by the degrees of freedom.

VI. LEXICAL AMBIGUITY AND PROBABILISTIC GRAMMARS

Grammatical ambiguity is unpleasant in that it generates numerical problems that have no nice solution, but at least grammatical ambiguity represents a conceptually clear problem. We have a sentence-type, and there are two or more trees for it. The case of lexical ambiguity is more puzzling.

Let the lexical form of a given sentence be the result of substituting the dictionary classifications for the words in the sentence. A word is lexically ambiguous if the classification for that word represents two or more grammatical categories. (See Chapter 3.) A lexical form is a terminal form (or, alternatively, a sentence type) only if there are no lexically ambiguous words in the original sentence. In allowing the multiple classifications of words in the dictionary, I created the situation of never being quite certain as to what terminal form a given utterance had. For example, 93 sentences in ERICA had the lexical form

*) pron#aux,pron#link art n.

This lexical form could represent either of the terminal forms

*)' pron aux art n

or

*)'' pron link art n .

Lexically ambiguous forms, such as *), can be thought of as a kind of shorthand, useful for a programmer but conceptually baggage that needs removal. Terminal forms such as *)' and *)'' use only symbols in the grammar GE1, while *) cannot have a probability according to GE1 without an explicit way of treating lexically ambiguous forms in a sample.

Since the dictionary introduces lexical ambiguity, it is appropriate to ask what is the dictionary's status in the analysis. One view is that the dictionary is a computational way of handling what is in fact a very large grammar. In *), the symbol

pron#aux,pron#link

is the lexical classification given to such a contraction as the word

that's .

If we adopt seriously the view that the dictionary is a "programmer's fiction", then we need to replace the dictionary with the underlying grammar upon which the analysis rests. This grammar would include a rule like

n -> boy

for a word such as

boy

that is classed as a noun (the symbol 'n'). For the word 'that's' we could include rules like

```
pron#aux -> that's
pron#link -> that's .
```

Actually, this is not quite context-free; however, we can remove contractions as we scan for words (the algorithm for which is representable by a finite automaton since it need only look at some fixed number of characters at one time--perhaps three.) Then, add such context-free rules as

```
pron -> that
aux -> is
link -> is .
```

An advantage of this method is that the terminals of the grammar are actually words rather than symbols standing for classes of words. Moreover, what I call 'lexical ambiguities' would actually be grammatical ambiguities, and hence according to this super-grammar, sentences could have well-defined probabilities. But the astounding grammar this would generate would have over 4,000 rules for ERICA, and likewise, the full-parameter model of the probabilistic grammar would have 4,000 independent variables. This would so dilute the evidence of the data that we would have no probabilistic theory left, and all but a few cliché-utterances would have negligible probability, even if I had the computational energy available, which I haven't. The use of the dictionary moves the theory-testing up a level of generality, from actual utterances to lexical forms of utterances. Abandoning the dictionary, I should have to predict the occurrence of individual words, and there simply is not enough evidence to do this (6).

There is a deeper reason than practicality for keeping the lexicon. I cannot believe that the simple parsing of simple sentences requires of a child the kind of computational energy that would be required of a computer to handle a 4,000-rule context-free grammar. My experience with parsers, both in connection with this work and in relation to systems programming, strongly suggests that this "brute-force" approach is not at all plausible. Hence, I am prone to believe that a lexicon plays an important theoretical role not to be subsumed by a grammar as such. This is another manifestation of the computation-performance orientation taken in this work.

---

(6) The large [K-F] corpus, referred to in Chapter 2, had over 1,000,000 word tokens in the sample. Even so, the frequencies given for many words are very likely not representative of written English.

As an example of the theoretical role that I think of the dictionary as playing, consider the classic ambiguous sentence:

*) I like flying planes.

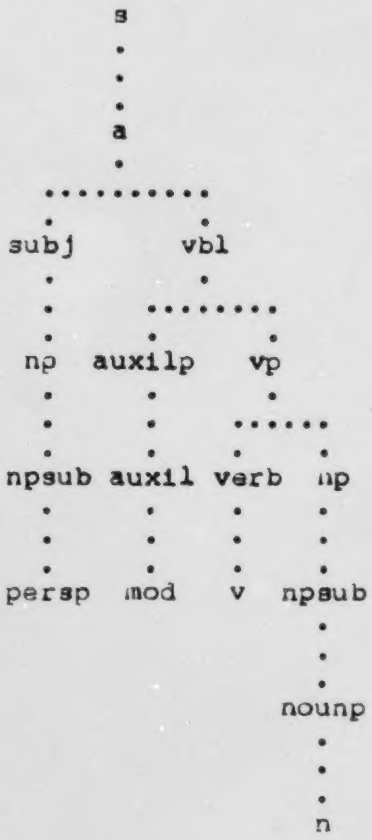
The ambiguity is of course whether the speaker likes to fly planes or likes planes that fly. I would assign to *) the lexical form

*)′ persp mod,v adj,v n .

Of the four alternative terminal forms represented by *)′, only two are parsed by the grammar GE1, and each of these corresponds to one of the expected ambiguities. Note that the other two alternative forms were rejected by GE1 as being ungrammatical. Here are the trees, as generated by grammar GE1.

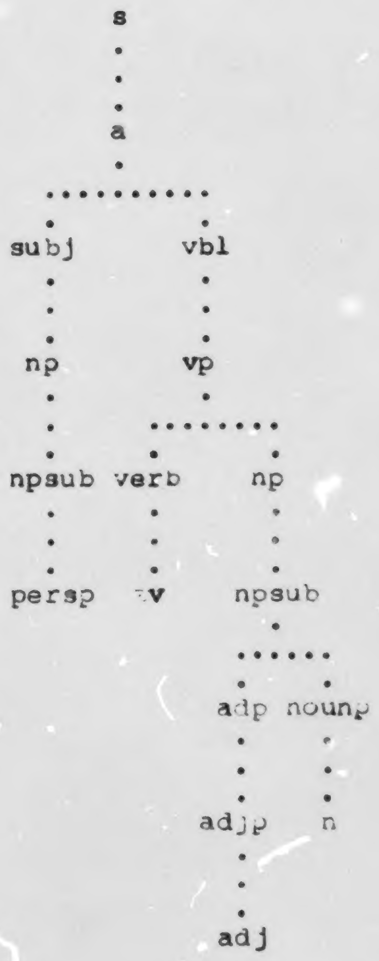
DERIVATION OF PERSP MOD V N

BY GRAMMAR GE1


s
a
subj
vbl
np
auxilp
vp
npsub
auxil
verb
np
persp
mod
v
npsub
nounp
n

DERIVATION OF PERSP V ADJ N

BY GRAMMAR GE1


s
a
subj
vbl
np
vp
npsub
verb
np
persp
v
npsub
adp
nounp
adjp
n
adj

The ambiguity of *) is lexical according to GE1 since the ambiguity is totally dependent upon the classification of the words in *). A view of how the hearer processes and responds to this sentence that is consistent with my work is that he first looks up the words in his dictionary (perhaps really a pre-selected subdictionary dependent upon the context), and then parses the resulting terminal form according to some grammar. Thus, which of the ambiguities I select depends on whether I "see" flying as an adjective or a verb. When the initial selection gets me into some kind of difficulty, I return to the lexicon for a subtler analysis of the words in the sentence.

For my purposes, I used three techniques to eliminate the lexical ambiguities present in ERICA. These methods are described below.

A. SPLIT THE PROBABILITY

The first thing that I tried was to divide up the observed frequency among the lexical and grammatical ambiguities. This method was an extension of the equal weights approximation for grammatical ambiguity.

Splitting the probability between lexical ambiguities corresponds to the assumption that the dictionary plays no theoretical role. Since I believe this is false, the method is a purely ad hoc way to get a meaningful probability distribution. I will describe it since I think it is an alternative that has to be dispensed with in order to understand the importance of the lexicon.

Actually, there are two variants of this method. They are:

1) Let FREQ(k) be the frequency associated with k in S. Then, if k has n alternative forms, let the corrected observed frequency of each alternative form be FREQ(k)/n. This simply assumes that each alternative form is equally likely.

2) Let COUNT1(k,n1) be the number of derivations for each n1 alternative form. Then, let COUNT(k) be the sum over the n alternative forms of COUNT1(k,n1). The corrected observed of form n1 is then

$$\frac{\text{FREQ}(k) * \text{COUNT1}(k,n1)}{\text{COUNT}(k)}$$

Both versions of the probability-splitting method were used, but I do not report the results in detail.

B. RESCANNER METHOD

A second way of handling the problem is to devise an algorithm for looking at the lexical ambiguities and deciding how to handle them. One explanation for this method is that it would extend the "methods" of the grammar to cases formally beyond the grammar. This interpretation better fits the probabilistic method (C below). What I have in mind in the rescanner model is something else.

The theoretical hypothesis I have in mind is that the initial response to a sentence consists of putting the sentence into a lexical form, _including initial disambiguation_, then proceeding to parse the terminal form or forms. If the sentence has a clear ambiguity (such as in many jokes, where the clear point is to have an apparent ambiguity as the basis of the humor), then the lexical form will be ambiguous; however, the listener will usually select the most likely classification _from the lexicon alone_ for the first pass at parsing the sentence. In the above 'flying planes' example, the listener might classify the word 'flying' as a verb before the parsing algorithm was even called. This method of lexical disambiguation is specifically oriented toward the listener.

C. PROBABILISTIC MODEL

The most satisfactory method of lexical disambiguation I have implemented is based on the _probabilistic model_. Briefly, each of the lexical ambiguities is assigned a probability, and the most likely ambiguity selected. The exact details of this approach are given below, after a discussion of the grammar GE1.

In the 'flying planes' sentence above, the alternative form persp mod v n

had probability .0014, and was hence selected by the model over the form

persp v adj n

which had probability .00016. The grammar would therefore select the reading of the sentence which means that the speaker likes to fly planes.

I am not personally convinced that this is the correct approach to lexical ambiguity. Particularly, I think that ambiguity is really semantical; but this does not preclude the possibility that disambiguation is done on the basis of syntax alone. I assume that the full machinery of language processing is seldom called into play.

However, the probabilistic model does one thing: it provides a concrete example of the meaningful use of a probability measure on a context-free grammar.

VII. THE GRAMMAR GE1

As mentioned, Table 3 contains the grammar GE1. This grammar is something of a compromise as it was developed from the interacting tension of four criteria, which are:

1. recognize as much of ERICA as possible;
2. minimize both grammatical and lexical ambiguity;
3. provide a good probabilistic model for the sample ERICA;

and, most importantly,

4. provide a good test for the semantical theory I had in mind.

Better grammars could no doubt be written for any one single purpose. Rather than include a whole complement of grammars in this work, I decided to include one that tried to be a complete model. I am pessimistic about the future of probabilistic grammars unless they are implemented in the service of disambiguation and semantical evaluation. Needless to say, grammar GE1 is the product of many dozens of discarded grammars.

Several high-frequency lexical forms are casualties of GE1, and are not recognized at all by the grammar. Appendix 5 lists those forms with frequency greater than or equal to 5, and shows: i) how many lexical ambiguities were in a form; ii) how many trees per lexical ambiguity;

iii) and the forms with frequency >= 5 that are not recognized by GE1.

Some of the high-frequency failures of GE1 are (7):

1) 28 adj adv .

Adding the rule

s -> adj adv

will of course parse this terminal form and will do so without affecting the rest of the grammar at all. There is, however, little to be gained by such an ad hoc solution; indeed, adding one rule to recognize one sentence-type is something of a loss. Of course, any corpus of n utterances can trivially be recognized by a cfg with n rules, so it is not surprising that a single rule can often be trivially added to a grammar.

2) 26 mod persp v,mod prep,adv adv
   10 persp v,mod prep,adv adv

Many of the forms not recognized represent a complex verb phrase, perhaps including modal verbs, prepositions, and adverbs. My efforts to include these in L(GE1) resulted in many added grammatical ambiguities elsewhere. A minimal distinction required to deal with verb phrases more adequately is the transitive-intransitive distinction in verbs.

---

(7) It is my practice to precede utterances, words, and phrases with a number. That number is the frequency in the data under consideration, usually the ERICA corpus.

The transitive-intransitive distinction is designed to distinguish between verbs that take no objects, and verbs that can take, say, a direct object. Unfortunately, the same verb can take 0, 1, or 2 objects (and perhaps more). Consider the uses of the verb 'to read' in the three sentences:

1) John is reading.
2) John is reading the Bible.
3) John is reading the Bible to a blind man.

Each sentence clearly uses the same word in (approximately) the same sense; yet the number of objects varies. If the constructions possible by the grammar depend upon the number of objects the verb may take, then we need to list 'to read' as several different kinds of verbs for usages that are not very different. Moreover, semantically there is no reason to stop at two objects--we might add object "slots" for time, place, and other adverbial concepts. In Chapter 5 I argue that the simplest semantical interpretation for verbs does not seem to require the transitive-intransitive distinction as a part of the syntax.

To carry out the transitive-intransitive distinction in a semantically sensible way would be to let "transitive" refer to verbs that may take objects optionally. This approach would, however, lead to classifying the objective cases of certain pronouns in the dictionary. (For example, the objective case of 'I' is 'me'.) My dictionary is not this subtle.

3) 13 persp aux,persp link qu,pron v

can be handled by adding the rule

persp aux pron v

to GE1. I did not do this because I am confused by the order of the verb in the sentence, and I also feel that I need the transitive-intransitive distinction to handle this.

VIII. LEXICAL AMBIGUITY IN THE ERICA CORPUS

Of course it is desirable to write a grammar that has a minimum of ambiguity, both lexical and grammatical. A cfg G can resolve a lexically ambiguous form if and only if exactly 1 of the terminal forms is recognized by GE1 . (If none at all is recognized, then the sense of resolution is that of dissolution, suitable for philosophers but unsettling to programmers.) The sentence

93 pron#aux,pron#link art n

is a case of resolution. The alternative terminal form

pron link art n

is recognized by GE1, while

pron aux art n

is not recognized by GE1. This is intuitively satisfactory if one looks at the 93 original sentences in the original corpus. When G resolves a lexically ambiguous lexical form, the alternative terminal form that was recognized is called the resolved lexical form. In the above,

pron link art n

is the resolved lexical form.

A slightly more subtle example of the resolution of lexical ambiguity occurs in the lexical form

*) 27 adv persp link,aux

where the alternative form

adv persp aux

is recognized while

adv persp link

is not. This is again intuitively satisfactory if we look at the actual 27 utterances in their original contexts; the reason is that adverbs seldom modify the linking verb.

Words classified as

link,aux

are the forms of the verb 'to be'. The reason for having a multiple dictionary classification for these words is that it is necessary to distinguish semantically their uses.

If k is a lexically ambiguous form with $n > 2$ alternative terminal forms, then G is said to reduce k if G recognizes n' of the n alternative forms, for $1 < n' < n$. Reduction may generate a new lexical form. When it does, the new form is called the reduced lexical form.

There is a great deal of lexical ambiguity in ERICA. Of the 2,995 types, 2,185 are lexically ambiguous. Many of the low-frequency sentence-types contribute to this pessimistic figure, since of the 9,085 sentence-tokens, only 4,419 are lexically ambiguous.

GE1 parses about 78 percent of the tokens in ERICA, and resolves about 56 percent of the lexical ambiguities. Table 5 details these results, showing both absolute numbers and percentages.

As a measure of the success of GE1 in removing lexical ambiguity, I calculated the ambiguity factor thus defined: for each sentence-type k in the sample, multiply FREQ(k) by the number of alternative terminal forms less 1. Then the ambiguity factor is the sum of this quantity over the k in the sample. The measure is intended to suggest how many "extra" lexical interpretations there are. The ambiguity factor for the complete corpus was originally 11,685; for that portion of the corpus parsed by GE1, the factor was 6,010, indicating that many very ambiguous sentence-types were not recognized by GE1; the ambiguity factor for the set of resolved and reduced lexical forms was 781. I take this to be quite an improvement, although the only data I have to compare it against are the results of (many) earlier grammars. One earlier grammar had had somewhat better values; however, it only recognized about 73 percent of ERICA.

TABLE 5

LEXICAL AMBIGUITY AND GRAMMAR GE1

CHILD PORTION OF ERICA

	TYPES	TOKENS
TOTAL SIZE	2,995	9,085
LEXICALLY AMBIGUOUS PORTION	2,185 72.95%	4,419 48.64%
NON-L.A. PORTION	810 27.05%	4,666 51.36%
PORTION PARSED BY GE1	1,394 46.54%	7,046 77.56%
PORTION OF L.A. PARSED	1,033 47.28%	3,030 68.57
PORTION OF NON-L.A. PARSED	361 44.57%	4,016 86.07%
L.A. COMPLETELY RESOLVED BY GE1	831 38.03%	2,464 55.76%
L.A. REDUCED BUT NOT RESOLVED	105 4.81%	194 4.39%

The resolution and reduction of lexical ambiguity reshapes the lexical forms present in the corpus, as originally distinct forms become the same. For example,

400 persp v#neg,mod#neg v

merges with two other forms to become

402 persp mod neg v

when the resolution of lexical ambiguity occurs. This merging process I call consolidation. GE1 recognized 1,394 of the original 2,995 types in ERICA. After consolidation, 1,125 types remained, still accounting for 7,046 tokens. This is encouraging since it means that there were fewer types in the sample than the original pass at the dictionary would have suggested.

The major onus (as far as this chapter is concerned) for accounting for the remaining lexical ambiguities comes from the need to obtain a sample that can have a probability distribution generated by a context-free grammar. Trying to resolve all such ambiguity by a grammar is an idea that is seductively difficult.

What is more possible is to devise an algorithm, perhaps with some context-sensitive elements, that extends the way that the grammar handles ambiguities when it is successful to the cases where it is not successful. This approach suggests a model with a rescanner that looks at unresolved ambiguities after an initial parse by a context-free grammar.

The "rescanner model" I used on the ERICA corpus simply picks the most "likely" single classification, in most cases. I looked at the ways in which GE1 resolved ambiguities, the frequencies of single classifications in the dictionary, and also the sentences themselves in developing the algorithm, which is shown in Table 6. The left-hand column is the ambiguous classification; the right-hand column shows what it was resolved to, and, in a few cases, gives a simple conditional rule.

TABLE 6

RESCANNER MODEL FOR DISAMBIGUATION

ALGORITHM FOR RESOLUTION OF LEXICAL AMBIGUITY REMAINING AFTER GE1

LEXICALLY AMBIGUITY	RESOLUTION
qu,pron	qu
n,adj	n
v,mod	v
v,aux	v
link,aux	link
persp,pronadj	pronadj
n,adv	n
v#neg,mod#neg	v neg
padj,pn#aux,pn#link	padj
padj,pn#aux,n#link	padj
persp#link,persp#aux	persp link
pron#aux,pron#link	pron link
persp#aux,persp#link	persp link
inter#aux,inter#link	inter link
aux#neg,link#neg	link neg
padj,pn#link	padj
prep,conj	conj
padj,n#link	padj
n#aux,n#link	n link
prep,adv	(if the next word or last word was adv, then prep, else adv)
n,v	(if n leaves the sentence all nouns, then v, else n)

The algorithm favors nouns, then adjectives, then verbs over the other classes. There is something vaguely to be said for the claim that this algorithm extends the methods of GE1. An exception is the resolution of 'qu,pron' to 'qu'. GE1 usually resolves to 'pron', since it does not leave a quantifier that modifies no noun phrase. The above algorithm, however, resolves 'qu,pron' to 'pron', since most of the remaining ambiguities are what appear to be noun phrases. The problem is caused by the rules that allow multiple noun-phrases to be noun-phrases; inadvertently, these rules let 'qu,pron' be either a 'qu', modifying the noun, or a 'pron', a part of a multiple noun-phrase. Two high-frequency sentences displaying this problem are

11 persp v qu,pron n
and
6 persp v prep qu,pron n .

The trees for these sentences are given in Table 7, thus illustrating the problem with multiple noun-phrases.

TABLE 7

TREES SHOWING CONFUSION IN GE1 OVER qu,pron

TREES FOR persp v qu,pron n

```
       s
       .
       .
       .
       a
       .
  ............
  .          .
subj        vbl
  .          .
  .          .
  .          .
 np         vp
  .          .
  .     ...........
  .     .    .    .
npsub verb  np   np
  .     .    .    .
  .     .    .    .
  .     .    .    .
persp   v  npsub npsub
             .    .
             .    .
             .    .
           nounp nounp
             .    .
             .    .
             .    .
           pron   n
```

```
      s
      .
      .
      .
      a
      .
  ..........
  .        .
subj      vbl
  .        .
  .        .
  .        .
 np       vp
  .        .
  .     ........
  .     .      .
npsub verb    np
  .     .      .
  .     .      .
  .     .      .
persp   v    npsub
               .
             ......
             .    .
          quart nounp
             .    .
             .    .
             .    .
            qu    n
```

TREES FOR persp v prep qu,pron n

```
        s
        .
        .
        .
        a
        .
  ..............
  .            .
subj          vbl
  .            .
  .            .
  .            .
 np           vp
  .            .
  .     ................
  .     .      .       .
npsub verb   prepp    np
  .     .      .       .
  .     .    ......    .
  .     .    .    .    .
persp   v  prep  np   npsub
                  .    .
                  .    .
                  .    .
                npsub nounp
                  .    .
                  .    .
                  .    .
                nounp  n
                  .
                  .
                  .
                pron
```

```
       s
       .
       .
       .
       a
       .
  ...........
  .         .
subj       vbl
  .         .
  .         .
  .         .
 np        vp
  .         .
  .     .........
  .     .       .
npsub verb    prepp
  .     .       .
  .     .    ........
  .     .    .      .
persp   v  prep    np
                    .
                    .
                    .
                  npsub
                    .
                  ......
                  .    .
                quart nounp
                  .    .
                  .    .
                  .    .
                 qu    n
```

Table 8 gives the statistical results of using the above rescanner model for disambiguation on ERICA, for the various combinations of the full parameter model versus the geometric model, and the chi-square versus the modified chi-square. All models group for expected frequency less than 5, and include the correction for continuity of .5, as explained above. The results are summarized only, and give the chi-square (or modified chi-square), the degrees of freedom, the chi-square divided by the degrees of freedom, and a statistic called the residual. The residual is simply the difference between the sum of the observed frequencies and the sum of the expected frequencies. It is therefore the number of sentences that the grammar predicted that we would find, for sentence-types that were not found at all. Recall that every sentence in L(GE1) has a non-zero probability, and that L(GE1) is infinite, since it contains some recursive rules. Hence, we should always expect a non-zero residual; but the smaller, the better. The size of the residual is yet another gauge of the goodness of fit.

TABLE 8

RESCANNER MODEL OF LEXICAL DISAMBIGUATION

PROBABILISTIC MODELS OF ERICA SPEECH*

GRAMMAR GE1

	MODEL CHI-SQUARE	RESIDUAL	DEGREES OF FREEDOM	GROUPS	CHI-SQUARE / DEGREES OF FREEDOM
Full parameter Chi-square	24,001.52	2,117.40	106	88	226.43
Geometric Chi-square	47,139.22	1,540.84	120	69	392.83
Full parameter Modified Chi-square	21,078.16	2,117.40	106	88	198.85
Geometric Modified Chi-square	14,219.49	1,540.84	120	69	118.50

---

* After consolidation, the rescanner model had 1,072 sentence types, still accounting for 7,046 tokens.

The most accurate method I used for lexical disambiguation is the probabilistic method. Starting with values for each b[i,j] , I computed the probability of each alternative lexical form for a type, and then selected the most probable alternative. (I used the values generated by the rescanner model given above as the parameters.) The method turned out to be uncannily subtle.

For example, on the lexical form

11 persp v qu,pron n

discussed above, the alternative

persp v qu n

had a probability of .0036 while the other alternative

persp v pron n

had only .00005. Likewise, for the form

persp v prep qu,pron n

the probability was .0000815 for

persp v prep qu n

which was preferred to

persp v prep pron n

with a probability of .0000119.

Of course the rescanner model made the same choices in these cases. The probabilistic model turned out to be much more sensitive in cases such as

3 qu,pron qu,pron qu,pron qu,pron qu,pron pron

Of the 32 alternative forms here, 13 were recognized by GE1. The rescanner model chose

qu qu qu qu qu pron

(which may well be correct) while the probabilistic model selected

qu pron qu pron qu pron

indicating, at least, that it is trying to follow the grammar closely.

Since the rescanner model always replaces 'qu,pron' by 'qu', in particular the lexical form

qu,pron

is resolved by the rescanner to

qu .

This is clearly unsatisfactory. The probabilistic model makes the intuitively correct choice, as is shown in Table 9, which includes the resolutions made by the probabilistic model where FREQ(k) >= 5.

After disambiguation by the probabilistic method, there were 1,060 types remaining (having begun with 1,125.) Table 10 gives the statistical results of the various ways of testing the fit.

Grammatical ambiguity remaining in the corpus is actually rather small. This could be because many of the classical "ambiguities" are lexical in nature. The following gives the number of types (and tokens) with various numbers of derivations. (A type has 1 derivation just in case it is not ambiguous.)

GRAMMATICAL AMBIGUITY REMAINING AFTER LEXICAL DISAMBIGUATION
PROBABILISTIC MODEL OF DISAMBIGUATION

NUMBER OF DERIVATION(S)	TYPES	TOKENS
1	980	6,919
2	78	125
3	1	1
4	0	0
5	1	1

About 92 percent of the types (98 percent of the tokens) in this reformed sample are grammatically unambiguous. This is sufficient, I claim, for assurance that the equal weights approximation method will give reasonable values to the maximum likelihood problem.

TABLE 9

PROBABILISTIC MODEL OF LEXICAL DISAMBIGUATION

SOME HIGH-FREQUENCY DISAMBIGUATIONS*

FREQ	RESOLUTION	COUNT	SOURCE	PROB
87	pron	(1,1)	qu,pron	.0135879
30	qu pron	(1,1)	qu,pron pron	.0012678
27	qu n	(1,1)	qu,pron n	.0033439
24	adv persp mod	(1,1)	adv persp v,mod	.0005833
14	v qu n	(1,1)	v qu,pron n	.0008767
12	persp v qu n	(1,1)	persp v qu,pron n	.0003635
11	persp v	(1,1)	persp v,mod	.0161562
9	inter link adv adv	(1,1)	inter#aux,inter#link adv adv	.0001362
8	persp mod neg	(1,1)	persp v#neg,mod#neg	.0007625
7	persp link	(1,1)	persp link,aux	.0013975
7	mod neg persp	(1,1)	v#neg,mod#neg persp	.0003895
6	aff persp v	(1,1)	aff persp v,aux	.0001036
6	persp v	(1,1)	persp v,aux	.0161562
6	persp v prep qu n	(1,1)	persp v prep qu,pron n	.0000815
6	pron link pron	(1,1)	pron#link qu,pron	.0004769
6	pron link qu n	(1,1)	pron#link qu,pron n	.0001174
6	pron qu pron	(1,0,1,1)	pron,qu qu,pron pron	.0000338
6	v persp	(1,1)	v,aux persp	.0116550
5	aff persp link	(1,1)	aff persp link,aux	.0000090
5	link pron art n	(1,1)	link,aux pron art n	.0000008
5	persp v qu pron	(1,1)	persp v qu,pron pron	.0001378
5	persp aux neg	(1,1)	persp aux#neg,link#neg	.0002396

---

*The SOURCE is the lexically ambiguous form. The numbers in the COUNT indicate, for each alternative form, the number of derivations of that alternative according to GE1. PROB is the probability associated by GE1 to the alternative that is best, which is then the RESOLUTION.

TABLE 10

PROBABILISTIC MODEL OF LEXICAL DISAMBIGUATION

PROBABILISTIC MODELS FOR THE GRAMMAR GE1

MODEL	CHI-SQUARE	RESIDUAL	DEGREES OF FREEDOM	GROUPS	CHI-SQUARE / DEGREES OF FREEDOM
Full parameter Chi-square	22,215	2,108	109	90	203.81
Geometric Chi-square	45,776	1,487	125	72	366.21
Full parameter Modified chi-square	15,834	2,108	109	90	145.27
Geometric Modified chi-square	12,206	1,487	125	72	97.65

Appendix 6 contains the complete printout of the b[i,j] 's for the full parameter and geometric models. Also, I include a run of the full parameter model on the sentence-types with frequency >= 5, which is Appendix 7. A complete printout of this would run several hundred pages.

IX. PROBABILISTIC GRAMMARS AND UTTERANCE LENGTH

In Appendix 1 I discuss the length of utterances in ERICA, and offer several probabilistic models to account for utterance generation. Table 3 of Appendix 1 gives the length distribution for the entire corpus, showing that the most probable length is 1, followed closely by 2 and 3. While the negative binomial distribution fits this reasonably well, as it stands it suggests no mechanism for utterance production.

A probabilistic grammar is such a mechanism. Given a (non-zero) distribution to a grammar G, each sentence in L(G) has a probability. Hence, for each length i, there is a probability associated with i, which is the sum over $k \in L(G)$ such that $|k| = i$ (8).

I have computed this sum for $i=1,\ldots,4$ (9). The results follow (using the parameters resulting from the probabilistic model of lexical disambiguation). Included also are the number of utterances in L(GE1) with a given length; this number grows surprisingly quickly.

---

(8) See [Suppes-2] pp. 25-29.

UTTERANCE LENGTH ANALYSIS

Length	Freq(in L(GE1))	Prob
1	17	.298
2	180	.238
3	1,242	.182
4	8,929	.135

no. of utterances = 10,368
total probability = .853
residual probability = .147

The first four lengths account for about 85 percent of the probability of utterance distribution. Using these values as a predictor for the values in Table 3 of Appendix 1, we find the following results.

---

(3) The algorithm I used for this computation is to generate all the length-i utterances (in internal representation in my programs) and check each one. Since there are 21 terminals in the grammar GE1, this means that the program had to check 204,204 possible utterances, which required 40 minutes of computation time! A much more efficient method would be to look "top-down" at the sentences, expanding the tree according to some strategy; however, the programming investment is beyond the worth of the question in connection with this work.

OBSERVED VS. PREDICTED UTTERANCE LENGTHS

GRAMMAR GE1

LENGTH	OBSERVED FREQ	THEOR. FREQ.	THEOR. PROB.	CHI-SQUARE
1	2,072	2,707.33	.298	149.09
2	2,064	2,162.23	.238	4.46
3	1,950	1,653.47	.182	53.18
4	1,142	1,226.47	.135	5.82
TOTAL	7,228	7,749.50	.853	212.55
PERCENT	.7959	.853		

GE1 predicts that we will find about 85 percent of the utterances in this range. In fact, only about 80 percent are there. I think that the explanation is that GE1 is simply incomplete, in that it doesn't parse as many of the more complicated forms as it should.

CHAPTER 5 -- SEMANTICS

I. METAMATHEMATICAL SYNTAX AND SEMANTICS

Model-theoretic semantics was invented by Alfred Tarski to make precise the notion of the meaning of a first-order sentence in terms of a set of objects D called the domain of the model, and a set of primitive relations and functions on the domain (1). The primitive terms of a first-order language are the variables and constants. It is convenient to allow that these denote individual objects in the domain. Complex terms and formulas then have their denotations defined recursively from the denotations of the simple terms and the rules of composition given in the language.

I offer the following simple example of a first-order language L1, with its truth definition. (There is, of course, nothing new in this treatment. I give it simply to provide continuity of notation.) The language is a fragment of quantifier-free arithmetic; for simplicity, I omit the quantifiers and variables they bind, and consider only a more restricted case.

---

(1) See "The Concept of Truth in Formalized Languages" in Logic, Semantics, and Metamathematics by Alfred Tarski.

The language L1:

constant terms: a , b

function symbol: +, a two-place operator

predicate symbol: =, a two-place predicate

parentheses: (,) to show grouping

logical connectives: =>, v, &, ¬

1. The set T of terms contains the constant terms, and if x,y are in T, then (x + y) is in T. Nothing else is in T.

2. The set F of formulas contains:

   i) if $x, y \in T$ then $(x = y) \in F$;

   ii) if $a, b \in F$ then
   $$\begin{array}{ll} (a \Rightarrow b) & \in F \\ (a \vee b) & \in F \\ (a \,\&\, b) & \in F \\ (\neg a) & \in F \end{array}$$

   iii) Nothing else is in F.

The intended model for L1 is the domain D of the positive integers, where the symbol + means addition, the symbol = means equality of two integers, the constant a denotes 0, and the constant b denotes 1. Note that the domain satisfies the familiar property of closure, whereby if i, j are in D, then the sum of i and j is also in D. This is necessary since all of these sums represent terms in the language, and each term must denote.

I now give, informally, the rules for the meanings of the formulas in F. Notice that each rule corresponds to a way or process by which formulas are created.

i) $(x = y)$ is true just in case the denotation of x is identical to the denotation of y;

ii) $(a => b)$ is true just in case if a is true, then b is true;

iii) $(a \vee b)$ is true just in case a is true or b is true;

iv) $(a \& b)$ is true just in case a is true and b is true;

v) $(\neg a)$ is true just in case a is false.

We can now show that each formula of F is either true or false under the model provided, and it is clear that the interpretation is "intuitively satisfactory" -- i.e., the "true" formulas correspond to well-known truths of arithmetic.

The above interpretation for $L1$ is deceptively satisfying. Nothing about the syntax requires that this, the intended interpretation, be the only one. In particular, we have stated no axioms to even guarantee that such properties as commutativity or transitivity apply to the function symbol $+$. A primary goal of model theory is to characterize, given a language, the classes of models that various sets of sentences in the language can have. In order to do this it is necessary to characterize the notion of a model.

The characterization is that of a relational *structure*. Let

$$\mathfrak{A} = \langle D, P1, \ldots, Pn, F1, \ldots, Fm, a1, \ldots, ak \rangle$$

(where l,m,n are natural numbers)

be a *relational structure* if and only if

i) D is a non-empty set of objects;

ii) for each Pi, i=1,,,n, there is an ri, called the *rank of Pi*, such that

$$Pi \subseteq D^{ri} \quad ;$$

iii) for each Fi, i=1,,,m, there is an ri, again called the *rank of Fi*, such that

$$Fi: D^{ri} \longrightarrow D$$

(i.e., Fi is a function on D^{ri} into D) ;

iv) each ai, i=1,...,k, is an element of D.

Following this definition, the class of models for the language L1 is any structure

$$\mathfrak{A} = \langle D, F, A, B \rangle$$

where D is nonempty, F is a function on D^{2} into D, and A,B are elements of D.

It is not enough to give a model $\mathfrak{A}$ for L1; it is also necessary to show how *valuations* for each $f \in L1$ are constructed. This is done by associating *semantical rules*, in the form of set-theoretical functions, with the rules of formation for the formulas of L1.

VALUATION OF TERMS:

i) basis conditions

$$[a] = A$$
$$[b] = B$$

([a], or more explicity $[a]_{\mathfrak{A}}$, means the *valuation of a in* $\mathfrak{A}$.)

ii) recursion condition

$$[(x + y)] = F([x],[y]).$$

VALUATION OF FORMULAS:

i) basis condition

[(x = y)] = if [x] = [y] then true else false.

ii) recursion conditions

[(x => y)] = if [x] is false or [y] is true then true else false

[(x v y)] = if [x] is true or [y] is true then true else false

[(x & y)] = if [x] is true and [y] is true then true else false

[(¬x)] = if [x] is true then false else true

There is an important distinction to be made between three kinds of symbols in the language. Some symbols -- a,b,+ -- denote objects in the model $\mathfrak{A}$; these I call *denoting symbols*. Other symbols -- =>,v,&,¬,= -- signal the use of certain semantic rules, such as implication or identity, but do not denote objects in $\mathfrak{M}$. These I call logical symbols. Finally, parentheses (and sometimes commas, brackets, and braces) make grouping clear. These I call utility symbols. Utility symbols may be eliminated from first-order logic by using polish notation, wherein the order is implicit.

II. CONTEXT-FREE AND METAMATHEMATICAL SYNTAX

The treatment of the language L1 given in Section I corresponds in style to that usually encountered in logic textbooks. It is worth noting, given the convention of using generative grammars in linguistic studies, that there is a certain correspondence between the definition of syntactic classes by giving closure conditions of sets, as above, and the use of context-free grammars. The language L1 can, for example, be defined by the following cfg G, where

$$G = \langle V, T, F, P \rangle$$

$$V = \{ =, =>, v, \&, \neg, +, a, b, (,), T, F \}$$

$$T = V - \{T , F\}$$

and P contains the rules

(1,1) F -> (T = T)
(1,2) F -> (F => F)
(1,3) F -> (F v F)
(1,4) F -> (F & F)
(1,5) F -> (¬ F)

(2,1) T -> a
(2,2) T -> b
(2,3) T -> (T + T)

Then, the semantic rules associated with the closure conditions can be associated instead with the productions of G , mutatis mutandis.

It is of some interest to ask what the relation is between context-free grammars and the kinds of definitions obtained by giving closure conditions on classes, since the former is standard in linguistics while the latter is used extensively as the syntactical basis for model theory. The usual requirement for logical syntax is that the sets must be recursive, and there are recursive sets that are not context-free. However, the full complement of recursive methods is not needed for the fundamental syntactic notions of the formal languages of mathematical logic; several such syntactic classes are usually defined by a kind of closure that I call simple closure. It is necessary to formalize this notion of simple closure, as a kind of syntactic meta-meta theory of mathematical logic.

NOTATION: a,b,c are syntactic objects;
S, T, U are sets of syntactic objects;
x, y, z are syntactic variables
ranging over sets of syntactic
objects.

The following are primitives:

a set M of symbols;

an operation & on symbols in M,
known as concatenation (2).

a symbol mem denoting membership--
e.g., a mem S;

the symbol then denoting a conditional.

the symbol and denoting a conjunction.

Syntactic Objects (S.O.)

i) $M \subseteq$ S.O.—i.e., symbols are syntactic objects;

ii) if $\alpha, \beta \in$ S.O., then $\alpha \,\&\, \beta \in$ S.O.

S.O. corresponds to the class T+ associated with context-free grammars.

---

(2) The set M corresponds to the terminal vocabulary T of a cfg G. However, the operation for a grammar corresponding to concatenation is to put a space or a plus sign between two symbols being "concatenated". Concatenation is intuititively putting symbols side by side; but the grammarian does not write

ADJN

but rather

ADJ N

or

ADJ + N

The problem is one of notation, hinging on the difference between a "symbol" as a formal object and a "symbol" as a typographical character.

Syntactic Terms (S.T.)

i) S.O. $\subseteq$ S.T.;

ii) if x is a syntactic variable then $x \in$ S.T.;

iii) if $\alpha, \beta \in$ S.T. then $\alpha \,\&\, \beta \in$ S.T.

S.T. corresponds to the class V+ associated with a cfg.

Positive Boolean Expressions (P.B.E)

i) if x is a syntactic variable and S is a set, then

x mem S

is a P.B.E.;

ii) if $\Gamma_1, \Gamma_2 \in$ P.B.E., and no syntactic variable occurring in Γ_1 occurs in Γ_2 or conversely, then

Γ_1 and $\Gamma_2 \in$ P.B.E.

Simple Closure Conditions (S.C.C.)

i) if $\alpha \in$ S.O. then

α mem S

is an S.C.C. (on S);

ii) if $\Gamma \in$ P.B.E., $\alpha \in$ S.T., then

Γ then α mem S

is an S.C.C. (on S).

iii) the extremal clause ("Nothing else is in S.") is an S.C.C. (on S).

S is defined by simple closure iff only finitely many S.C.C. define S. S1, S2, ,..., Sn may be defined simultaneously provided there are no infinitely descanding sequences of definition.

Theorem 1. The class of simple-closure definable S is equivalent to the class of context-free languages.

I indicate the proof by giving the algorithms for generating a set of S.C.C. given a cfg, and conversely.

Proof.

1) CFG => S.C.C. Suppose we have a grammar

$G = \langle V, T, S_G, P\rangle$.

Then, let

$M = T$.

We are to define, by S.C.C., the class S corresponding to L(G).

First, rewrite G into equivalent Chomsky normal form

$G' = \langle V', T, S_G, P'\rangle$.

Each rule in P' is of the form

i) A -> a

or

ii) A -> B C

where A,B,C are non-terminals, and a is a terminal. (See Chapter 3.) For each rule of the form i), use the S.C.C.

a mem A ;

for each rule of the form ii), use the S.C.C.

(x mem B) and (y mem C) then $x \alpha y$ mem A.

It is clear that S = L(G).

2) S.C.C. => CFG

Suppose S is defined by simple closure.

Then we need a grammar

$G = \langle V, T, S_G, P \rangle$.

Let

T = M

and let S_G be a symbol corresponding to the class S defined by simple closure. Then, if

α mem A

is an S.C.C. on A, for $\alpha \in$ S.O., then let

$A \to \alpha$

be in P. Since α is an S.O., it is a non-empty string of symbols in M.

Suppose $\Gamma \in$ P.B.E. , $\alpha \in$ S.T., and

Γ then α mem A

is an S.C.C. Then we reduce this according to the rules for P.B.E. If Γ is of the form

x mem B ,

then replace occurrences of x in α by B and call the result of this replacement α [B substituted for x]. Then, add to P the rule

$$A \rightarrow \alpha \text{ [B substituted for x]} .$$

If Γ is of the form

Γ_1 and Γ_2

then perform any such replacements of the variables in Γ_1 and Γ_2 into the variables in α. Notice that, since rule ii) for P.B.E. requires that no syntactic variable in Γ_1 occur in Γ_2 (and conversely), there will be no problem in making this substitution. Now, add to P the following rule:

$$A \rightarrow \alpha \text{ [correct variable substitutions]} .$$

It is clear that the above translation will, with the appropriate proofs by induction, yield the actual proof of the theorem.

Many of the elementary syntactical notions of the first-order predicate logic can be defined by simple closure; hence, by the above translation, an equivalent context-free grammar can be obtained. The sets of variables, predicates, terms, and well-formed formulas are examples. In practice it is customary to assume an infinite class of variables, and since the above formalization of S.C.C. allows only a finite class of symbols, some way of generating the variables, e.g. using prime symbols, is necessary. The following defines the class of variables VAR, assuming primitive symbols v and '.

i) v mem VAR;

ii) x mem VAR then x' mem VAR.

iii) Nothing else is in VAR .

Infinitely many constants, as well as infinitely many predicates and functions of arbitrary type, can be generated by similar devices.

While the set of well-formed formulas WFF is defined by S.C.C. and is hence a cfl, the set of formulas of a first-order language, STCE, is not definable by S.C.C. (3). Also, the class TAUT of tautologies is not a cfl. (Obviously, the class LT of theorems of first-order logic cannot be a cfl since, by Church's theorem, that class is not even recursive. It is less obvious that recursive classes, such as the class of tautologies, is not a cfl.)

The results that STCE and TAUT are not cfl can be proven by use of a result known as the "uvwxy theorem".

Theorem 2 (the "uvwxy theorem"):

---

(3) A sentence is a formula with no free occurrences of variables, where an occurrence is free if it is in the scope of no quantifier binding that variable.

Let L be any cfl. Then, there exist constants p,q depending only on L such that if there is a word z in L, with $|z| > p$ (where $|z|$ is the number of symbols in z), then z may be written as $z = uvwxy$, where $|vwx| <= q$, and v and x are not both ε (the empty symbol) such that for each integer $i >= 0$,

$$uv^{i}wx^{i}y$$

is in L (4).

This theorem limits the amount of context checking that a cfg can perform; intuitively, it says that a finite number of sentences can be checked for context, but an effort to check several contexts over an infinite class of sentences will result in some extraneous strings being accepted by the grammar. The theorem makes it explicit how to find such extraneous sentences.

I will indicate how Theorem 2 is used by proving the following result.

Theorem 3: The set of sentences of a first-order language with a single monadic predicate P is not a cfl.

Proof.

---

(4) For a proof of the uvwxy theorem, see [Hopcroft-Ullman], pp. 51-52.

Suppose to the contrary that STCE of the language with one monadic predicate is a cfl. Then, for each natural number j, the formula

$$z_j \; : \; \forall v'^{j} (P (v'^{j}) \to P (v'^{j}))$$

is in L , since these are closed WFF 's. Let p,q be the constants guaranteed by the "uvwxy" theorem. Then, select a j such that

$$|z_j| > p$$

and

$$| \forall v'^{j} \; | > q \, .$$

Clearly, j is a simple function of p,q; further, z_j is in L.

This satisfies the hypotheses of the "uvwxy" theorem, so we know that we can rewrite z_j as uvwxy such that v and x are not both empty, and for each $i >= 0$,

$$uv^{i}wx^{i}y \quad \text{is in SCTE} \; .$$

The key is to show that any way of dividing zj (using linear notation for z sub j) into segments u,v,w,x,y will not avoid the extraneous introduction of some non-sentence into STCE. A counterexample to the proof would be one (nonempty) subsequence of zj that could be repeated indefinitely without generating a non-sentence, or a pair of subsequences that can be repeated together. The subsequence consisting of the quantifier and its variable could be repeated indefinitely and still yield an STCE; however, j was chosen so that the length of the quantifier and its variable would be larger than q, so this subsequence will not satisfy the hypotheses of the theorem. The only other repeatable subsequences are the strings of primes that make variables. Picking just one such subsequence will clearly cause non-sentences to be introduced. We can pick two such subsequences, repeating them together, such as the following division would indicate:

$$\underbrace{\forall v\,'^{\,j-1}}_{u}\underbrace{{}'}_{v}\underbrace{(\,P\,(\,v\,'^{\,j-1}}_{w}\underbrace{{}'}_{x}\underbrace{)\rightarrow P\,(\,v\,'^{\,j}\,)\,)}_{y}$$

But then

$$\forall v\,'^{\,j+1}\,(\,P(v\,'^{\,j+1}\,)\rightarrow P\,(\,v\,'^{\,j}\,)\,)$$

is in STCE, and it is a nonsentence.

The "uvwxy" theorem illustrates the following point: two counters (such as the counters on the number of primes) can be kept together by a cfg. But, if there are three or more counters, then each pair must be kept together by a different process in the grammar, and hence some extraneous results are unavoidable.

Notice that the set

$$\{ \forall v'^{j} P(v'^{j}) \mid i >= 0 \}$$

is a cfl; the appropriate grammar, with F as the start symbol, contains the productions:

(1,1) F -> ∀vA)
(2,1) A -> P(v
(2,2) A -> 'A' .

It is interesting to note that this grammar bears little relation to the semantics likely to be given to the formulas in question.

While the closed formulas of first-order logic do not form a cfl, it is well to point out the sense in which this would not be a restriction on a semantical theory based on a context-free treatment of first-order logic. The class WFF is a cfl, and we can allow that open formulas are meaningful. The usual convention is to let an open formula be equivalent to its universal closure--i.e., the formula obtained by surrounding the given open formula by universal quantifiers for each variable occurring free in the formula. (At least one text, Introduction to Logic by P. Suppes, uses the analogous existential closure.)

However, there is a real sense in which Theorem 3 limits the power of any semantics based on context-free languages. The concept that I propose using is that of a context-free semantics: I shall say that a semantics defined on a language is context-free if it is computable by a push-down automaton. The idea is that we cannot first present a cfg G, give an arbitrary algorithm for computing the meaning of a sentence in L(G), and then claim that the semantics itself is "context-free" because G is. The first-order logic is such an example: a semantics on, say WFF, must contain an algorithm for determining what the free occurrences of variables in a formula are. This algorithm cannot be represented by a push-down automaton; if it could, we could write a cfg for STCE, which Theorem 3 claims we cannot do. Hence, the grammar underlying such a semantics for first-order logic must be context-sensitive.

I think it is important not to consider this a limitation on the whole approach given here. It is admitted that natural language, with or without complex mathematical expressions, is not context-free. This does not preclude that there are large and useful fragments that are context-free. Moreover, the experience gained from working with context-free grammars may be easily transferred to work with more powerful classes of grammars.

A valuable point is that first-order logic can be put into the framework of generative grammar at all. Theorem 1, while mathematically trivial, has a philosophically important message in the context of much current work in computational linguistics. As Suppes explains (5),

> A line of thought especially popular in the last few years is that the semantics of a natural language can be reduced to the semantics of first-order logic. The central difficulty with this approach is that now as before how the semantics of the surface grammar is to be formulated is still unclear . . . how can explicit formal relations be established between first-order logic and the structure of natural languages?
> (emphasis added)

The difficulty of looking for first-order representations of natural language is not here considered to be that first-order logic is insufficiently expressive. As I have attempted to show, it is semantically more powerful than context-free grammars. I should be happy with any formal language representation of natural language (into even a programming language such as LISP or ALGOL) as long as there existed a powerful theoretical "translation" between the surface of natural language and the formal language. The superficial arguments for using set-language over first-order logic are those of custom dating back to Tarski, of convenience, and the fact that first-order logic has its semantics given in terms of set-language. The deeper reason is that first-order logic can be defined by generative grammars (some concepts admittedly requiring context-sensitivity), and so we may think of the semantics for natural language, based on generative grammar, as being amenable to the set-theoretical approach that has been so successful for symbolic logic. An intermediate pass through first-order sentences does not appear to be a gain in clarity or concept.

---

(5) [Suppes-2], p. 1.

III. MODEL STRUCTURES AND CFG

The basic idea behind any semantics for a cfg is that the terminal symbols (tend to) denote set-theoretical objects in the model structure, and the rules of the grammar (tend to) be interpreted by set-theoretical functions. In practice, there is however a certain tradeoff between the denotations given to the terminals and the functions associated with the rules--i.e., which symbols are denotative and which are logical. It seems that a certain number of philosophical controversies have been engendered from this possibility of a tradeoff.

As an example, looking at the language L1 for a fragment of quantifier-free arithmetic (see Section I above), the following alternative model structure $\mathfrak{A}$ and set-theoretic rules can be given for L1.

$\mathfrak{A}$ = < D, PLUS, EQUAL, IMP, OR, AND, NOT, 0,
1, TRUE, FALSE>

where

i) D = w U { TRUE, FALSE }, where w is the set of natural numbers;

ii) PLUS is a function from D^2 into D (denoted PLUS: D^2 --> D);

iii) EQUAL: D^2 --> {TRUE, FALSE}

iv) IMP: {TRUE, FALSE}^2 --> {TRUE, FALSE}

similarly for OR , AND

v) NOT: {TRUE, FALSE} --> {TRUE, FALSE}

vi) 0,1,TRUE,FALSE $\in$ D.

The following are the denotation rules, assigning objects in $\mathfrak{A}$ to terminals in L. If α is a symbol or a sequence of symbols, let [α] be the denotation of α in $\mathfrak{A}$. Then,

[=] = EQUAL

[=>] = IMP

[&] = AND

$[v] = \text{OR}$

$[\neg] = \text{NOT}$

$[+] = \text{PLUS}$

$[a] = 0 \ , \ [b] = 1$

$[(] = 0 \ , \ [)] = 0.$

Finally, I give the functions corresponding to the rules of the cfg G that generates L1.

LABEL	RULE	FUNCTION
(1,1)	$F \rightarrow (T = T)$	$\{ b \mid (\exists \langle [T],[T],b \rangle \in [=] \}$
(1,2)	$F \rightarrow (F \Rightarrow F)$	$\{ b \mid (\exists \langle [F],[F],b \rangle \in [\Rightarrow] \}$
	for rules (1,3) , (1,4) , and (2,3) similar functions are required;	
(1,5)	$F \rightarrow (\neg F)$	$\{ b \mid (\exists \langle [F],b \rangle \in [\neg] \}$

The model $\mathfrak{A}$ is somewhat unusual. The point in its construction was to make every linguistically significant symbol have a denotation and to eliminate the notion of a 'logical' symbol. Even the parentheses "denote", but since they do not play a part in the set-theoretical functions it is of no consequence. (The use of parentheses is, of course, to avoid ambiguity.) All the work is done by the denotations given to the terminal symbols. The semantic functions simply say to apply the arguments in the appropriate manner, and thus have no real content.

While it may seem arbitrary whether this model is used, or the more usual one given in Section I above, the question of which symbols denote objects is a key disputation in much philosophical work. The Frege-Russell tradition of the ontological status of propositions is based in, or at least permitted by, the formal plausibility of objects in a model that behave like propositional functions. As is well known, paradoxes creep into somewhat richer languages than L1 when semantical notions such as 'true' and 'false' are given ontological status. One solution is type theory with its hierarchy of propositional functions; but this is beyond the limits of my discussion. Without committing myself to any position whatever regarding the status of propositions, the formal fact remains that there is an interplay between the denotations of the terminal symbols and the set-theoretical functions associated with the rules of the grammar.

As an example of the problem I would like to avoid, consider the noun-phrase

*) capitol of France.

*) contains a prepositional phrase. A reasonable way to interpret prepositions is as some kind of function. Consider however two alternative grammars and the semantics they offer for *).

G1: (1,1) NP -> NP of NP

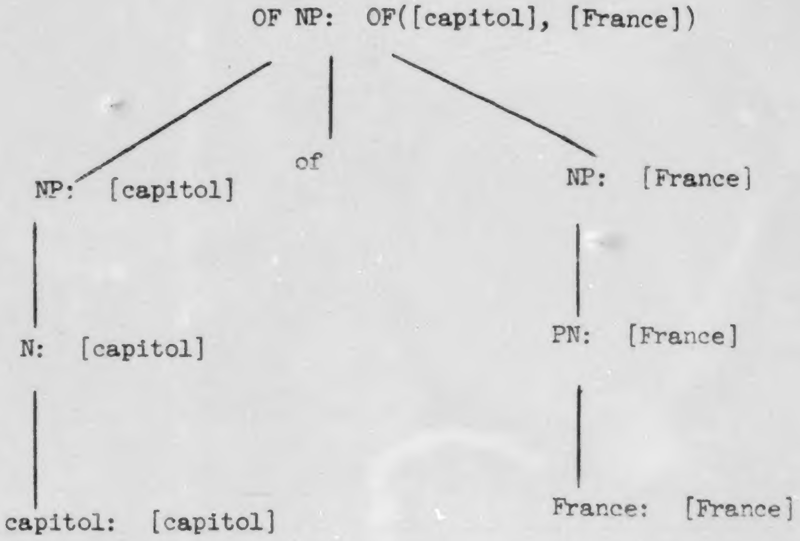
(1,2) NP -> N

(1,3) NP -> PN

plus, of course, the appropriate lexicon. The semantic functions corresponding to the rules of G1 are:

(1,1) OF([NP],[NP])
(1,2) the identity function
(1,3) the identity function

Then, the semantic tree for *) is:


OF NP: OF([capitol], [France])
of
NP: [capitol]
NP: [France]
N: [capitol]
PN: [France]
capitol: [capitol]
France: [France]

Notice that the word 'of' does not denote; instead the rule (1,1) assumes a rather dubious set-theoretical function. 'Of' is a logical symbol in the grammar. An alternative grammar G2 has a denotation [of] in the model. The rules of G2 are:

(1,1) NP -> NP PREP NP
(1,2) NP -> N
(1,3) NP -> PN

with the appropriate lexicon; the semantical rules are:

(1,1) $[NP] \cap \{ a \mid (\exists \langle a,b \rangle \in [PREP]) (b \in [NP]) \}$
(1,2) identity
(1,3) identity

G2 is to be preferred to G1 in that it makes clear a kind of ontological commitment: namely, that the information about the function associated with the preposition 'of' has to be a part of the model structure (which is, in relation to Erica's linguistic behavior, the data base) and cannot be considered a part of the set-theoretical functions available (which correspond to the machinery of language processing) (6). It is my belief that much of the talk about the ontological commitment of natural languages would benefit from an understanding of this kind of a tradeoff.

Further, I think that this appears to run contrary to much of the talk about the 'logic' of various words. It seems to me that much of the talk about, say, the way in which modal notions ('believe','know') are used has suffered from too little empirical evidence. Hence, if I am uncertain about how a word functions semantically, I prefer to make a commitment to an object in the data base representing that word, in the hope of collecting some hard data on the use of the word. This puts the emphasis upon understanding linguistic behavior rather than analyzing concepts, but only because I think the former has been overlooked. In the case of modal concepts, a more complicated structure is needed than the one I have given for ERICA; I have tried to consider only the extensional case, leaving modal notions as transparent. Readers familiar with Kripke-Hintikka-Montague semantics for modal

---

(6) There is a better way of handling many prepositions, such as 'of' and 'with', and that is to create a function by combining the preposition with a phrase. In *), the appropriate combination is

capitol of

and the commitment is to a function on D mapping objects (countries) into their capitols and giving some kind of error condition (say, by returning the null set as the capitol of non-countries).

In any actual implementation of a data base, I think this kind of approach would be necessary in order to give a reasonable structure to the data. I have not used this approach here, because I am simply too awash in data already.

notions in modal logic (best thought of as an extension of first-order predicate logic) will realize that the possibility exists of giving more complex set-theoretical structures.

IV. SEMANTICS FOR ERICA

The model theory of the classical first-order logic requires only a simple model-theoretical structure containing objects in the domain, and n-ary relations and functions on the domain. Natural languages require more complicated structures than first-order languages. Following Suppes (7) I give the closure conditions defining the class $H(D)$, based on a domain D. This will allow for any finite composition of functions in the natural hierarchy of sets but may be stronger than any application requires.

Let D be a nonempty set. (In general, D may be finite, for my purposes.) Then I define $H(D)$ to be the smallest set such that:

i) for each $n \in \omega$ (the set of natural numbers), $D^n \in H'(D)$;

ii) if $A, B \in H'(D)$, then $A \cup B \in H'(D)$;

---

(7) See [Suppes-2], pp. 10-14.

iii) if $A \in H'(D)$, then $P(A)$, the power set of A, is in $H'(D)$;

iv) if $A \in H'(D)$ and $B \subseteq A$, then $B \in H'(D)$.

The denotation of a true sentence will be a special object TRUE, and likewise a false sentence denotes the object FALSE. I let

$$H''(D) = H'(D) \cup \{\text{TRUE},\text{FALSE}\} .$$

Since some utterances will in fact express two "propositions" (see below), we need to allow ordered pairs of denotations. Hence, let

$$H(D) = H''(D) \cup \{ \langle x,y\rangle \mid x,y \in H''(D) \} .$$

Set-theoretical functions are now associated with the rules of a cfg. Let $G = \langle V, T, S, P \rangle$ be a cfg, and Φ a function on P that assigns to each $p \in P$ exactly one set-theoretical function such that if the right-hand side of p has n symbols, then $\Phi(p)$ has n arguments. The arguments are to be applied to $\Phi(p)$ in the same order as they occur in the rhs of p (8). Then $G' = \langle V, T, P, S, \Phi \rangle$ is a *potentially denoting* cfg.

Notice that no rule can have more than one semantical function associated with it. Should I want a grammatical construction to have two or more semantic interpretations, I would proliferate rules in the grammar accordingly rather than associate more than one function with a rule. Since a derivation is associated with a tree (see Chapter 4), this means that if a sentence is semantically ambiguous, then it is syntactically ambiguous as well. It seems desirable to mirror semantic ambiguity in syntactic ambiguity so that if a terminal-form is semantically ambiguous (i.e., has two or more interpretations that are not set-theoretically equivalent), it is grammatically ambiguous as well.

---

(8) The explanation for the order of arguments requirement is to provide a fiat solution to a problem mentioned in [Suppes-2]. The problem can be summarized by noting that two or more instances of the same symbol may occur at different nodes of a tree and will generally play non-interchangeable roles in the semantics of the sentence. To avoid labeling trees and reformulating the definition of a derivation accordingly, I simply require that the symbols on the rhs of a production p have their valuations applied in order to the set-theoretical function associated with p. This creates rather strange functions (such as converse subset), which I ignore by using the standard set-theoretical terminology as metalinguistic abbreviations, assuming that all is clear. In any case, the program that I wrote to do the work knows what is happening, but it is of no conceptual interest to go through the thrashing of explicit definition on this point.

The convention I use for my abbreviations is this: if a symbol occurs two or more times in a string, then the valuation of the string is written using the symbols with subscripts that refer to the order in the original string; if the order of the symbols in the valuation is the same as the order in the string, then the subscripts are omitted. For example, I write:

[N LINK N] = if [N] $\subseteq$ [N] then TRUE else FALSE.

The conditions on H(D) that allow ordered pairs of denotations need some explanation. Often, the most reasonable approach to the semantics of an utterance is to believe that it expresses two (or perhaps more) propositions. For example, consider the question

Did you go or did you stay?

Clearly, this is two separate questions. Answering 'yes' to the utterance (a favorite response of the logically sophomoric) misses both the intent of the questioner and the logic of the question. What is needed is something like an ordered pair with the elements corresponding to the two separate questions (9). For such utterances, it would not be satisfactory to suggest two alternative semantic analyses. The notion of alternative implies that, while we have two or more possibilities, only one is correct and to be acted upon. The idea here is rather that the utterance conveys two separate packages of information.

In the grammar GE1, there are five rules that have associated functions using ordered pairs of denotations, rules (8,4), (8,5), (8,6), (8,11), and (8,15). Table 1 gives the terminal forms using each rule. It is most plausible that rules (8,11) and (8,15) should not be generating a pair of denotations, since there is evidence in the ERICA corpus that the utterances these terminal-forms represent are simply repetitions. However, I have left these rules in the grammar since it is the more general case.

The full generality of the closure conditions on $H(D)$ are not realized in ERICA, since the terminal-forms requiring paired denotations all have an affirming or negating word as one of the "propositions".

---

(9) A large part of the informal work that I did with the ERICA corpus concerns the question-answer pairs; it is from this subset of ERICA that the clearest view of the interaction between speakers arises, so I have asked if the semantics handles these interactions correctly. I plan a later paper on the semantics of questions with an attempt to predict the answers, syntactically and semantically. Unfortunately, the ERICA corpus is a little small for this analysis, but at IMSSS at Stanford we have a larger corpus that is being collected under conditions experimentally superior to those used in ERICA.

TABLE 1

TERMINAL-FORMS IN ERICA REQUIRING PAIRED DENOTATIONS

RESCANNER MODEL OF LEXICAL DISAMBIGUATION

RULE: (8,4)   s -> neg a

FREQ	TERMINAL-FORM
20	neg n
8	neg persp link neg
6	neg adj
6	neg pron link art n
5	neg art n
4	neg mod persp v persp
4	neg n n
3	neg adv
3	neg persp v neg
3	neg v
2	neg adj adj
2	neg pron link n
2	neg persp link n
2	neg persp link art n
2	neg pron link art adj n
2	neg persp mod neg v prep
2	neg qu n
1	neg adj n
1	neg adv adj
1	neg art n conj art n
1	neg mod persp
1	neg mod persp v pron
1	neg mod persp v prep persp n
1	neg n v
1	neg n pn
1	neg n v neg
1	neg n pn v prep pronadj n
1	neg pn
1	neg prep qu n
1	neg persp v n
1	neg pron link
1	neg prep persp
1	neg prep padj n
1	neg persp v pron
1	neg persp v art n
1	neg persp v adj n

```
1        neg persp link pn
1        neg persp v persp
1        neg persp mod neg
1        neg prep pronadj n
1        neg pron conj pron
1        neg pron link pn n
1        neg persp link adj
1        neg pronadj n aux v
1        neg persp aux neg v
1        neg persp mod v persp
1        neg pron link pronadj
1        neg persp v persp pron
1        neg persp link neg adj
1        neg persp link art pn n
1        neg pron link neg art pn
1        neg persp link art adj n
1        neg persp mod neg v pron
1        neg persp aux v prep persp
1        neg pron link pronadj adj n
1        neg persp mod v prep pronadj n
1        neg persp mod neg v pron prep pron      *
1        neg qu
1        neg v n
1        neg v pron
1        neg v persp prep
TYPES = 61     TOKENS = 120

STARRED FORMS HAD TWO TREES, WITH EACH TREE
USING THIS RULE ONCE.


                   RULE: (8,5)     s -> aff a


11       aff persp v
9        aff persp mod
5        aff persp link
1        aff mod persp n
1        aff n
1        aff pron link
1        aff persp link adj
1        aff persp link art n
1        aff persp mod v persp
1        aff prep n prep persp
TYPES = 10     TOKENS = 32


                   RULE: (8,6)     s -> a aff
```

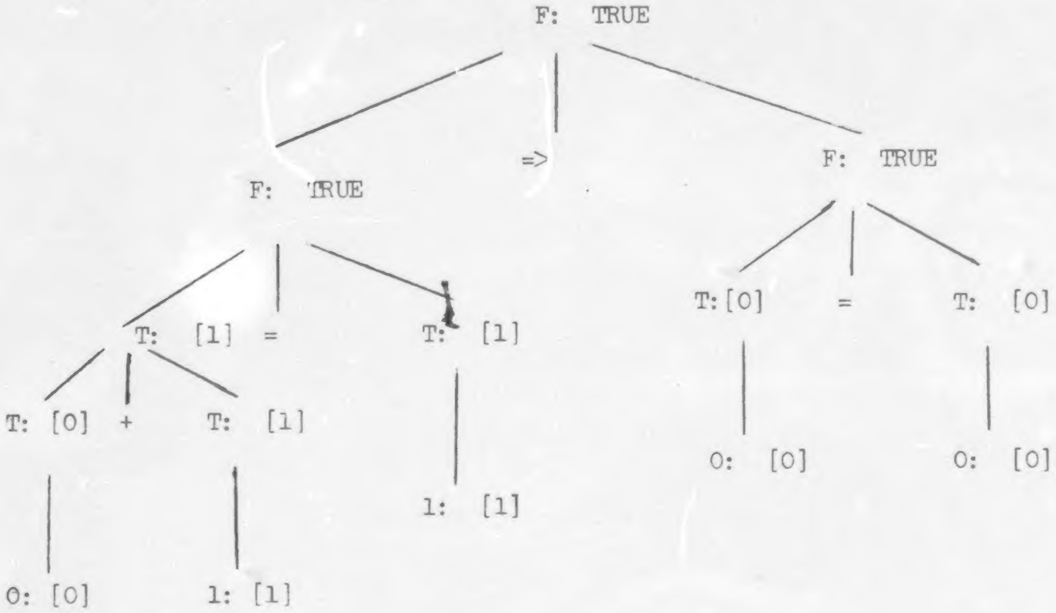
```
1       persp mod neg v n aff
1       v aff
TYPES = 2    TOKENS = 2


                  RULE:  (8,11)  s -> aff aff


42      aff aff
TYPES = 1    TOKENS = 42


                  RULE:  (8,15)  s -> neg neg


5       neg neg
TYPES = 1    TOKENS = 5
```

Usually the basis for the recursion into $H(D)$ is provided by a function v on the set of terminals T. If $\alpha \in V+$, let $v(\alpha)$ be denoted by $[\alpha]$, as an abbreviation. Thus, terminals denote.

Strings of terminals and nonterminals "denote" in the sense that the basis denotations of terminals together with the semantical rules on the grammar generate a valuation. For example, in the language L1 , the formula

*) $(((0+1) = 1) => (0 = 0))$

"denotes" its truth-value (TRUE), determined by following the semantic tree for *).



I shall write, again as an abbreviation,

$$[((((0+1) = 1) \Rightarrow (0 = 0))] = \text{TRUE} .$$

There is, however, a distinction that should be made here, namely, between a denotation made on a (string of) symbol(s) by a basis assignment, as opposed to the valuations generated by the rules of the grammar. I say that the former is a *basis valuation*. If the basis valuations on a potentially denoting grammar G into a model $\mathfrak{A}$ are all on the terminals of G, then $\mathfrak{A}$ is said to be a *uniform model for G*.

My model for the semantics of ERICA is expressly not uniform, since I wish to make some basis valuations on two terminals. The problem arises with verbs that take prepositions as a part of the verb itself, especially where the verb may be separated from the preposition.

Let t1, t2 $\in$ T. Then t1#t2 means the string consisting of t1 and t2, with # acting as a space marker. For some such combinations of terminals, there is a basis denotation. Such terminals are the separable verbs together with their associated prepositions. Without requiring that the parser be context-sensitive, special set-theoretical functions associate with the rules that generate the terminal forms where these separable verbs occur. Such a set-theoretical function is a *non-uniform function*. In the grammar GE1 (see Chapter 4), there are two rules using non-uniform functions, (3,8) and (4,35), each having its own associated function. Table 2 lists the terminal forms (from the rescanner lexical disambiguation model) that require these rules. Each terminal-form in Table 2 is grammatically unambiguous relative to GE1.

TABLE 2

SENTENCES GENERATED BY RULES REQUIRING NON-UNIFORM FUNCTIONS

RULE: (3,8) vp -> verb np prep

14	persp v persp prep
12	persp v pronadj n prep
10	persp mod v persp prep
6	persp mod neg v persp prep
3	n v persp prep
3	persp v art n prep
3	persp mod v pron prep
3	persp mod neg v pronadj n prep
2	mod persp v persp prep
2	persp v n prep
2	persp v pron prep
2	persp mod neg v art n prep
1	art n v persp prep
1	conj persp v art n prep
1	conj persp v pronadj n prep
1	conj persp mod v art n prep
1	conj persp mod neg v persp prep
1	int pn v pronadj n prep
1	int persp mod v persp prep
1	n mod v persp prep
1	n persp mod v persp prep
1	n v n prep
1	persp v qu n prep
1	persp mod v art n prep
1	persp mod neg v n prep
1	persp mod v pronadj n prep
1	pn v n prep
1	pn v persp prep

TYPES = 28 TOKENS = 78

RULE: (4,35) a -> vol subj prep

34	v persp prep
5	v pron prep
4	v art n prep
3	v pronadj n prep

```
1        int v persp prep
1        mod neg v pronadj n prep
1        neg v persp prep
1        v n prep
1        v pn prep
1        v prep pronadj n prep
1        v qu n prep

TYPES = 11    TOKENS = 53
```

There are in ERICA 39 types representing 131 tokens that require that two terminals have a basis valuation together. Non-uniformity of a model $\mathfrak{M}$ could of course account for the phenomenon of attributivity, such as the phrase "alleged dictator", but I don't find any great need for this in the ERICA corpus.

V. SEMANTICS FOR GE1

Most of the lexical categories given in the dictionary have a specified kind of valuation in H(D). Since I have tried to use simple semantic functions for ERICA, a certain complexity is placed upon the basis valuations of the terminals. I think this is desirable because it makes an explicit commitment to the information that is in the "data base" (Erica's perception, her memory, the physical surroundings of the conversation). Also, it gives us a feel for the adequacy of simple functions for the semantics of natural language.

Of course, I cannot give the basis valuations of the individual words, as they would be spelled out in a data base dealing with a specific subject matter. Rather, for each grammatical category, I can indicate the kind of object in the structure H(D) that is appropriate.

A. NOUNS, PRONOUNS, AND ADJECTIVES

The following grammatical categories have simply subsets of the domain as their basis valuation:

adj
n
padj
persp
pn
pron
pronadj

These are the nouns, pronouns, and adjectives. Some words, such as proper nouns, denote one object. Thus, the word

Erica

just refers to the person Erica. By fiat, the denotation [Erica] of the word 'Erica' will be a singleton set containing the element

Erica.

This should cause no confusion. With this convention, the semantics is simplified in that the denotation of a noun or proper noun will always be a set of objects; the semantical functions assume this.

This group dominates the corpus. Looking back to the data on dictionary construction, Table 9 of Chapter 3 shows the (relative) numbers of words with the various lexical classifications. I summarize that data below:

WORDS THAT TAKE SUBSETS OF THE DOMAIN

AS THEIR CLASSIFICATION (10)

(ADJ, N, PADJ, PERSP, PN. PRON, PRONADJ)

	ENTIRE CORPUS	ADULT PORTION	ERICA PORTION
TOTAL TYPES	3,490	3,135	2,039
TAKING SUBSET	2,411	2,169	1,389
PERCENT	69%	69%	68%

Hence, by types, 68 percent of the words in ERICA take the subset denotatation according to this model.

3. VERBS

There are four kinds of verbs in the ERICA lexicon:

aux
mod
v

plus the forms of 'to be' that are classed as

link.

There is an important semantical difference between the forms of 'to be' and other verbs; I discuss the other verbs first.

The problem with verbs is that they take objects.

---

(10) These, and other figures of this kind, are computed from Table 9 of Chapter 3. When a contraction is encountered in that table, if one of the symbols in the contraction is the desired symbol it is added in.

More importantly, the same verb will sometimes take 0, 1, or 2 objects. Consider the (fictitious) examples:

i) I am reading.
ii) John is reading the book.
iii) Mary is reading a blind man the Bible.

One semantic approach is to view i) and ii) as elliptical, in which case, the semantics might have to account for the suppressed arguments to the [read] predicate.

An approach that makes less commitment in this direction is to let the semantics of a verb be of the form

$A \cup B \cup C$,

where $A \subseteq D$, $B \subseteq D\hat{}2$, $C \subseteq D\hat{}3$,

and D is the domain of the model. A purely intransitive verb (e.g., 'to run') has $B=C=0$. A verb that always takes one object has $A=C=0$, $B \neq 0$. Most transitive verbs can have 1 or 2 objects, and in this case $A=0$, $B \neq 0$, $C \neq 0$. The more general case is of a mixture.

Again referring to Table 9 of Chapter 3, I give the sums of the types that have one of these three classifications:

aux
mod
v .

WORDS THAT ARE VERBS IN THE ERICA CORPUS DICTIONARY

(LEXICAL CLASSIFICATION aux OR mod OR v)

	ENTIRE CORPUS	ADULT	ERICA
TOTAL TYPES	3,490	3,135	2,039
TYPES AS VERBS	899	513	812
% AS VERBS	26%	25%	26%

It is possible to allow verbs to have a large number of objects, either explicitly or implicitly, indicating time, place, other personal objects. I have avoided this for the present.

Verbs classified as LINK (forms of 'to be') are not included in the above since I have considered them as logical symbols and used semantical rules accordingly. LINK in a terminal-form signals the use of the subset function. For example, the terminal-form

12 pron link n

has as its valuation

$$\text{IF } [\text{pron}] \subseteq [\text{n}] \text{ THEN TRUE ELSE FALSE },$$

and, likewise,

$$44@[\text{persp v pron}] =$$

$$\text{IF } [\text{persp}] \subseteq \{ a \mid (\exists \langle a,b \rangle \in [\text{v}])(b \in [\text{pron}]) \} \text{ THEN TRUE ELSE FALSE }.$$

(This notation is read "the terminal form 'persp v pron', with 44 occurrences, has as its valuation in $H(D)$ ...

".) Notice that, if 'persp' refers ('refers' used informally) to only one object, then allowing the denotation of 'persp' to be the singleton means that subset is still the correct semantical function.

C. QUANTIFIERS AND ARTICLES

The implementation of quantifiers and articles is certainly the most important part of the semantics to the philosophically inclined. In fact, it is my suspicion that a logician will judge a theory of the semantics of natural language most on the ability of that theory to handle and coordinate quantifiers.

My theory will not satisfy many in this regard. I have not tried to develop a theory that will account for much mathematical language at all. On the basis of Theorem 3, I suspect that context-sensitivity is needed for this.

The rules of the grammar GE1 that introduce quantifiers and articles into sentences make use of the semantic function QUANTIF. QUANTIF is a function of two arguments, which are:

1) the denotation of the article or quantifier;

2) the denotation of the phrase being modified.

For example, the rule

(17,5) npsub -> quart adjp nounp

introduces quantifiers and articles into noun phrases.

(See the grammar GE1 in Table 3 of Chapter 4.) The semantic function for this rule is

$$\text{QUANTIF}([\text{quart}],([\text{adjp}] \cap [\text{nounp}]))$$

(wherein we use the symbols on the right-hand side of the rule to indicate the application of arguments). The semantic function QUANTIF is defined in this section, and it depends not only on the denotations of the words, but also on the words themselves--i.e., which quantifier or article was present. However, the function QUANTIF is still a part of a context-free semantics, in that the valuation returned by QUANTIF does not depend upon the context of the phrase in the sentence.

I now indicate the denotations of the various quantifiers and articles, where applicable, and the algorithm for computing the function QUANTIF.

1. CARDINAL NUMBERS

Most of the cardinal numbers less than 20 occur in ERICA. (Recall that cardinal numbers are classed as 'qu'.) Most of the usages are trivial, as for example in counting exercises. I give cardinal numbers denotations reminiscent of the Frege-Russell treatment of the notion of cardinal, although simplified. The method is to let a cardinal n be the set of all sets of D of cardinality n. For example,

$$[\text{one}] = \{x \in P(D) \mid |x| = 1\}$$

$$[\text{two}] = \{x \in P(D) \mid |x| = 2\}$$

$$[\text{three}] = \{x \in P(D) \mid |x| = 3\}$$

Notice that no use is made here of any sort of hierarchy despite the fact that a more complex use of language than that found in ERICA might require it. Consider the sentence, "Two groups of girls were present." The reasonable denotation [two] would have to include the set

$$\{ x \mid (\exists y,z \in x)(y,z \in D) \wedge (\forall w \in x)(w=y \vee w=z) \}$$

When the quantifier is a cardinal number, the valuation given to QUANTIF is given by

$$\text{QUANTIF}([\text{cardinal number}],[\text{a string}]) = [\text{cardinal number}] \cap P([\text{a string}]).$$

Hence, for the phrase 'two pretty girls' we obtain

$$\text{QUANTIF}([\text{two}],[\text{pretty girls}]) = \text{QUANTIF}([\text{two}],([\text{pretty}] \cap [\text{girls}])) = [\text{two}] \cap P([\text{pretty}] \cap [\text{girls}]) .$$

This gives us the class of all two-element sets of pretty girls.

Such noun phrases as 'the two pretty girls' do not occur in ERICA; however, I indicate now to handle these phrases in the next section.

2. THE DEFINITE ARTICLE

The definite article, 'the', occurs at least once in 358 sentence types, representing 377 tokens, among the 9,085 tokens in ERICA. Uses of 'the' can be classed as *demonstrative* and *intensive*, where the former serves to distinguish an object while the latter seem to do little semantically at all. Some examples of the actual sentences follow.

DEMONSTRATIVE USES OF 'the'

FREQ	SENTENCE
3	to the zoo we went.
2	in the water.
2	in the castle.
2	put it on the microphone.

INTENSIVE USES OF 'the'

FREQ	SENTENCE
2	i lost the other one.
1	all the clothes.
1	and the soldiers will come.
1	all the shapes.

This distinction is certainly not hard and fast, but making it tends to point out the degrees of semantic import the word 'the' has.

In the classical theory of definite descriptions, the word 'the' is treated as an operator picking out the object uniquely possessing a certain property; the classical example is, of course

*) Scott is the author of Waverly.

where the phrase 'the author of Waverly' denotes Scott uniquely. The logical form (11) of this sentence is something like

s = (iota x) W(x)

where s is the constant denoting Scott, W(x) is the predicate for 'x wrote Waverly', and iota is the definite description operator.

Looking at the usages of the word 'the' in ERICA suggests a more complicated notion of description. Nearly 10 percent of the usages of 'the' occur with plural noun phrases, such as

The tapes are going around.

Plurality could, of course, be accommodated by picking out a distinguished set, which may have more than one element. The classical theory of definite description has usually been stated only for predicates that are true of one object, but the extension to sets is an obvious one.

---

(11) I am somewhat unhappy about using the phrase 'logical form', since it may evoke many things beyond what I intend. I use the notion informally to mean the sentence in first-order logic, with set notation, that would be the representation for the given English sentence. I have nothing more formal in mind than the talk about translating ordinary language that is customary in elementary logic courses.

My inspection of the uses of 'the' in ERICA leads me to believe:

1) it is clear that the phrases using 'the' are perfectly clear to Erica and her conversants, so nothing very strange is happening;

2) the word 'the' is doing something -- it has semantical import, and is not always there merely for some kind of syntactic filling, as I had suspected might be the case;

3) while 'the' is picking out a distinguished set of objects, it is not clear that many phrases might be simultaneously meaningful, such as:

the man
the two men
the five men
the three most handsome men

To countenance this in a theory that extends the classical theory of descriptions, I suggest the notion of contextual orderings.

The first semantical concept I offer is the notion of the set IMMED , the set of objects of immediate importance to Erica. The initial reason for offering this is that many of Erica's utterances are elliptical and assume a limited domain for much of the conversation. Of course, the conversation may gradually change in topic, and when it does, the domain of immediate importance will change. Language provides for ways of "changing the subject", for example, by using proper nouns to bring new objects to the forefront of the conversation.

The set IMMED is the contextual parameter in my semantic model that contains the things of contextual interest or concern to Erica. The assumption is that careful examination of the context of utterance, the physical surroundings, and the notes of the adults would enable us to estimate this parameter at any given time and to account for the ways that objects are added to and subtracted from IMMED. I think that it is not as large a set as one might suspect.

The need for a contextual parameter in the semantics is illustrated by looking at various phrases in ERICA and noticing that the same phrase will appear to denote different things in different occurrences of the phrase. Notice the occurrences of the noun phrase 'the water' in the following utterances from ERICA.

SOME OCCURRENCES OF THE PHRASE 'the water' IN ERICA

FREQ	UTTERANCE
3	in the water.
1	he goes in the water.
1	he spilled the water.
1	lookat the water.
1	that's the water and let me go in there.

Looking at the contexts, it is utterly implausible to believe that the same object is denoted throughout.

Hence, the need for a contextual parameter.

I will define IMMED from the set IMMED1. Let IMMED1 be a subset of the domain D. The interpretation is that the elements of IMMED1 are the objects of importance in the conversation (at a given time).

Let R be a binary relation (ordering) on the set IMMED1 satisfying the following properties:

i) TRANSITIVITY: if xRy and yRz then xRz, for $x,y,z \in$ IMMED1;

ii) CONNECTEDNESS: xRy or yRx, for $x,y \in$ IMMED1;

Thus, R is a weak ordering. One of the requirements, connectedness, may be too strong. Intuitively, x R y means 'x is at least as important as y'.

Based on the structure given to IMMED1 by the ordering R (which may present a lot of structure, or very little), I want to include certain subsets of IMMED1 in IMMED. Perhaps I can motivate this by the claim that I think the following phrases may all be meaningful:

the men
the man
the three men

while, at the same time,

the two men

may be meaningless, or at least sufficiently unclear as to require a 'HUH?' from the listener. My claim about a conversation, such as the ERICA corpus, is that at each moment in the conversation there exists a set of objects IMMED1 together with the relation R, which intuitively means the relative importance of the objects in IMMED1.

It is now possible to define IMMED from IMMED1 and R. Actually, I want to define IMMED relativized to some set T, so I define first the set IMMED(T). Then, IMMED = IMMED(D), where D is the domain.

Let IMMED(T) be the smallest set such that

1) $(\mathrm{IMMED1} \cap T) \subseteq \mathrm{IMMED}(T)$;

2) if $S \subseteq (\mathrm{IMMED1} \cap T)$ then $S \in \mathrm{IMMED}$ if and only if

$$(\forall x \in (\mathrm{IMMED1} \cap T) - S)\ (\forall y \in S)$$

$$[\text{if } xRy \text{ then not } yRx \text{ and if } yRx \text{ then not } xRy]\ .$$

I shall call such a set S a *clean section of IMMED1 relative to T*.

Thus, IMMED contains the objects of contextual importance IMMED1, together with those subsets of IMMED1 that can be determined by the ordering, subject to the requirement that a subset must be neatly delineated by the ordering.

It is now possible to give the algorithm for the semantic function QUANTIF in the case that the article is the word 'the'. It is a conditional expression, indicating a series of evaluations to be attempted.

*) QUANTIF([the],[<expression>]) =

1) if <expression> is syntactically singular,
and there is a singleton set S in
IMMED([<expression>]) such that S $\subseteq$ [<expression>],
then evaluate to: S;

ELSE

2) if <expression> is syntactically singular,
then there is no evaluation.

ELSE

3) if IMMED1 $\cap$ [<expression>] is not null
then evaluate to: IMMED1 $\cap$ [<expression>]

ELSE

4) if <expression> contains a cardinal number, let s be the size of the elements of [<expression>]; then [<expression>] is computed by

QUANTIF([<cardinal>],[<expression 2>])

for some <expression 2>. If there is a unique set S $\in$ IMMED([<expression>]) such that $|S| = s$ and S $\subseteq$ [<expression 2>], then evaluate to: S

ELSE

5) the expression *) does not evaluate.

As an example, consider the phrase

the five men.

Let IMMED1 = {a1,...,a15,t},

and [men] = {a1,...,a15,b,c,d},

and let the relation R be given by the following diagram (where the higher elements are more important, and elements on the same level are equally important.)

a1

a2 a3 a4

a5 a6 a7

a8 t

a9 a10 a11 a12

a13 a14 a15

We restrict the ordering to [men] ([five men] would be more correct--this would require some added complexity of the above conditional function). This removes the element t from consideration. The only 5- element clean section is the set

{a8,a9,a10,a11,a12} ,

and hence, that is the denotation of the phrase 'the five men'.

The phrase 'the men' denotes the set

{a1,..., a15} .

Since there is no 2-element clean section, the phrase 'the two men' does not denote. The phrase 'the man' selects two 1-element clean sections. The above algorithm says that it therefore does not denote. Alternatively, we might select the highest clean section, and let [the man] = {a1}

which is intuitively correct.

Notice that the algorithm gives the classical results of the theory of definite description where applicable, yet the theory is extended to include other sets as well that are a part of natural discourse.

3. THE INDEFINITE ARTICLE

When the quantification theory of predicate logic is applied informally to natural languages, the existential quantifier is often used to represent the indefinite articles 'a' and 'an'. These words occur somewhat more frequently in ERICA than the definite article.

INDEFINITE ARTICLES IN ERICA

	TYPES	TOKENS
a	788	857
an	15	16

These words modify singular noun phrases exclusively. Presumably,

[a] = [an] ,

so I will identify the two forms of the indefinite article and talk only about 'a'. In about one-third of the cases, 'a' points rather non-specifically, as if to say some singular but unidentified, perhaps unfamiliar, object. Such cases include:

1 there's a farmer in there.
1 those are for a boy.

In many other cases (perhaps as many as 500) the word 'a' functions as a kind of generic pointer, meaning "something of this kind or satisfying these properties". Examples of this include:

2 i want to read a book.
1 you are making a house.

When Erica says

i want to read a book.

it is plausible that she is thinking of the criteria that specify "bookness", rather than a class of books (12).

---

(12) The treatment of semantics herein considered is extentional. Without involving myself in a discussion of modalities *de dicto* and *de re*, I would like to remark that there is more than a little modality in Erica's speech.

One solution that has occurred to me--one that is reasonably consonant with set-theoretical semantics--is to have *essential objects* in the data structure (ontology, if you will). In this way, the denotation of the phrase

a book

could be an essential book. I am tempted to recommend this as an explanation for linguistic development of children. Perhaps there is a confusion between properties and objects, and the child, in learning a cluster of properties, reifies them. Or perhaps parents foster a realism upon the child (one that they themselves have discarded) to facilitate learning the difference between oranges and pears.

I think this is something to consider in examining the semantics of children's languages.

The most straightforward definition of QUANTIF, when the article is 'a', is

```
QUANTIF([a],[<expression>]) =
    IMMED ∩ [<expression>]
```

This seems to work rather well in cases where the article occurs in the predicate of the utterance. For example,

```
[i'm a big girl] =
        if [i] ⊆ (IMMED ∩ ([big] ∩ [girl]))
            then TRUE else FALSE .
```

The grammar GE1 is deficient in regard to the semantics of many phrases containing 'a'. There are approximately 100 utterances in ERICA that contain 'a' in the subject for which GE1, as it stands, gives the wrong semantics. Consider the utterance

1 a boy had that one.

The logical form of this utterance is something like

$$(\exists x)\ (x \text{ is a boy and } x \text{ had that one}).$$

The rules of GE1 simply check to see whether or not the subject is a subset of the predicate. Hence, we have

EVALUATION 1:

if [a boy]

(continued)

$$\subseteq \{\, x \mid \exists \langle x,y\rangle \in [\text{had}]\ (y \in ([\text{that}] \cap [\text{one}]))\}$$

then TRUE else FALSE.

Clearly, no denotation [a boy] makes this plausible. Instead, we need to change the rules for GE1 to check for 'a' in the subject, in which case we could have something like

EVALUATION 2:

$$\text{if } [\text{a boy}] \cap$$
$$\{\, x \mid (\exists \langle x,y\rangle \in [\text{had}])(y \in [\text{that}] \cap [\text{one}])\}$$
$$\neq 0 \quad \text{then TRUE else FALSE.}$$

Some additional rules (perhaps several dozen) need to be added to GE1 to generate sentences wherein the subject is modified by the indefinite article; the appropriate semantic functions can then be associated with these rules.

4. THE UNIVERSAL QUANTIFER

The word 'all' occurs in 100 utterance types, accounting for 128 tokens. For simplicity, I let

QUANTIF([all],[<expression>]) =
  [<expression>]

as opposed to, say, restricting [<expression>] to the set IMMED1. This appears to work in about 75 percent of the cases. The remaining 25 percent use the word 'all' in the sense of 'completely', as in

19[the kitty all green] =

if [the kitty] $\subseteq$ [green] then TRUE else FALSE .

This is rather strange; it says that the kitty is a green thing, rather than the stronger interpretation of being completely green. I take it that these cases use 'all' as an attributive adjective rather than a quantifier.

This use of 'all' occurs in ERICA only when <expression> is an adjective phrase, so the rules for QUANTIF could be modified if I were willing to handle attributive adjectives, which I am not. However, this would give the wrong result to

1) men are all mortal.

which presumably has the same meaning as

2) all men are mortal.

and therefore, 'all' is not attributive in 1).

Some utterances using 'all' follow.

6	all gone.
6	it's all gone.
4	he's all black.
4	it all gone.
4	they're all gone.
3	all finished.
3	that's all i got.
2	all up.
2	all i have.
2	he's not all black.
2	i all finished.
2	they're all gone.
1	it's all gone.
1	'cause they're all gone.

1 all well.
1 all gone?
1 all mine.
1 all gone ...
1 all the way.
1 all those ...
1 all fall down.

D. PREPOSITIONS

Prepositions are used in GE1 in two ways:

1) As a syntactic part of a verb associated with the preposition. Table 2 lists the sentence types requiring rules (3,8) and (4,35), which associate a verb with a preposition. It is important to realize that the semantic functions associated with these rules are not concerned with the denotations of the prepositions involved. For example, the lexical form

persp v pronadj n prep,adv

represents the utterances

4 i dumped my puzzles out.
1 i dump my puzzles out.
1 i put my dishes away.

The valuation of these is given by

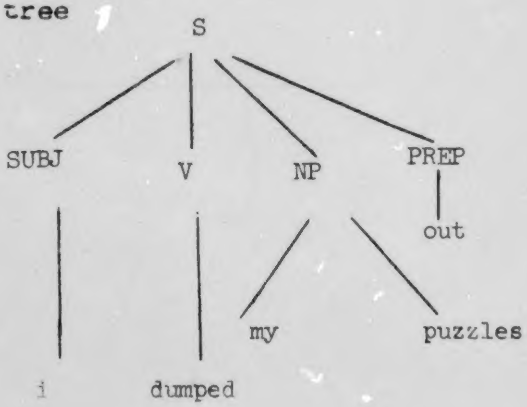
$$\text{if } [\text{persp}] \subseteq \{\, a \mid (\exists \langle a,b\rangle \in [\text{COMBINE}([v],\text{prep})])\ (y \in [\text{pronadj}] \cap [n]) \,\}$$

then TRUE else FALSE .

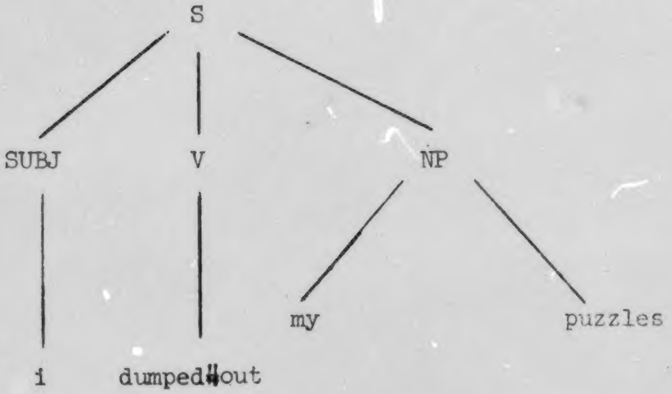
The syntactic function COMBINE concatenates the verb with the preposition to form, for example, the separable verb dumped#out .

This is then considered to be the syntactic unit in the utterance.

I might add that the function COMBINE does the same work that would be done by a transformation designed to convert the tree


S
SUBJ
V
NP
PREP
out
my
puzzles
i
dumped

to the tree


S
SUBJ
V
NP
my
puzzles
i
dumped#out

I do not explicitly use transformations; however, it might be clearer to do so in this case.

2) Two other rules, (7,1) and (8,2), allow prepositional phrases to modify noun phrases. (The reason for duplications of rules in the grammar GE1 relates to the fact that GE1 is also a probabilistic grammar. Often it is necessary to repeat the same process two or more times in a probabilistic grammar in order to account for statistical differences in the data.)

The denotation for a preposition is:

$$[\mathrm{prep}] \subseteq D\hat{}2$$

The rule that generates prepositional phrases is

(12,1) prepp -> prep np

and the semantics is

$$[\mathrm{prepp}] = [\mathrm{prep\ np}] = \{a \mid (\exists \langle a,b\rangle \in [\mathrm{prep}])\ (b \in [\mathrm{np}])\} \ .$$

Hence, the noun phrase

capitol of France

has as its denotation

$$[\mathrm{capitol}] \cap \{a \mid (\exists \langle a,b\rangle \in [\mathrm{of}])\ (b \in [\mathrm{France}])\} \ .$$

As previously mentioned, this is not the most natural way to handle prepositions. The preferable way is to view the preposition as a function--e.g.,

CAPITOL-OF(x) .

The preposition 'with' is perhaps a paradigm for my semantics for prepositions. In a quite natural way, [with] can be thought of as the set of pairs ⟨x,y⟩ such that x is in the accompaniment of y. Other prepositions, such as the ubiquitous 'of', do not in themselves represent a single, clear semantical notion, and hence my treatment does not do such prepositions justice.

E. ADVERBS

Adverbs form the most complex semantic class I've considered. Here I am particularly afraid that trying to make GE1 a good probabilistic grammar has hurt the semantic treatment.

Two views of the semantics of the adverb appear reasonable:

1) The adverb is a function. Given a set A, ADVERB(A) $\subseteq$ A, generally; for example, the adjectival phrase

[very good] = VERY([good])

where VERY is the function associated in the model $\mathfrak{A}$ with the adverb 'very'.

2) Alternatively, notice that most properties to which adverbs are applied can be thought of as orderings.

The adverb then selects the appropriate section of the ordering. As an illustration, suppose that the ordering given by the adjective ’good’ is:

ORDERING ON D GIVEN BY THE ADJECTIVE ’GOOD’

```
     |¯  x1
very |   x2 x3
     |_  x4
         x5 x6
         .
         .
         .
         x10
         x11 x12
         .
         .
         .
```

The adverb ’very’ then selects the appropriate part of the ordering in question.

I do not intend to develop either theory in any detail, except to remark that 1) seems a bit too general to be useful in analyzing a child’s language. 1) is a brute-force approach to the semantics of adverbs. 2) requires some analysis of the structure of some particular adjectives and adverbs in Erica’s speech, to see if it is tenable or not. (Incidentally, I think that the child thinks in terms of very clean and simple orderings on objects; I don’t think that the analysis of the ordering given by an adjective, say ’good’, would be as complicated as might be suspected.)

In the semantic functions I use the function

MEASURE of three arguments, which are:

1) The first argument is a dummy argument that preserves some of the structure of the subtree involved. It does not currently play a part in the semantics.

2) The adverb.

3) The set the adverb is functioning upon. Presumably, the concept represented by the set would have to provide an ordering. Hence, if 'pregnant' does not admit to "more and less", then 'very pregnant' is meaningless. (From experience, I am however quite certain that 'pregnant' does admit to degrees.)

Several rules--(4,21), (4,22), (4,23), and (4,38)--introduce interrogative adverbs (such as 'where', 'how') into the sentence. I now believe that these should be handled quite separately by a grammar with more individually suited rules.

F. OTHER WORDS

Interrogative pronouns (words classed as 'inter') ask questions. The meaning of a question Q, I shall say, is the set S such that a description of S is the correct answer to Q. Interrogative pronouns have no denotation, but are instead 'logical' words. (See Chapter 6 for a discussion of the rules that introduce interrogative pronouns.)

Other logical words include 'conj' (conjunctions) and 'neg' (negating words). Interjections ('int') play no semantic role in my analysis, either denotative or logical.

CHAPTER 6 -- THE SEMANTICS OF ERICA

I. THE SEMANTICS OF THE GRAMMAR GE1

In Chapter 5, I discussed the basis denotations given to the lexical categories of words in the dictionary. These denotations were, of course, selected with a mind to the kinds of semantic functions that would be assigned to the productions of the grammar GE1.

Here follows a discussion of the individual rules of GE1. For each rule, I give the semantic function, and then report on the results of using the rule on the data. Lexical disambiguation was accomplished by the probabilistic model of lexical disambiguation (see Chapter 4). In some of the more interesting cases, I list the terminal forms involved, and some of the original utterances (1). The format is the following: first the label and the production are given, then the following statistics about the usage of the rule in the ERICA corpus.

---

(1) I have tried to concentrate on the problems and inadequacies of this semantics in this section.

Space does not permit me to list all the transformations of the data that I used in preparing the summary given here, since it runs several thousand pages. However, the listings are available to anyone interested in this research in a more detailed way.

1) TYPES: the number of terminal forms that used the rule;

2) TOKENS: the number of original utterances that the TYPES represented;

3) TIMES USED: how many times the rule was used in ERICA (where a given terminal form may have used the rule more than once; this could either have been because one derivation of the form used the rule repeatedly or because there are several derivations of the form, each of which used the rule);

4) TIMES USED * FREQUENCY: the frequency of a form multiplied by the number of times the form was used, summed over the forms.

If the complete list of terminal forms is given for a rule, then the following information is included:

1) column 1: the frequency of the form in ERICA, after lexical disambiguation;

2) column 2: the number of derivations of the form by GE1;

3) column 3: the form, followed by the number of times the rule was used for the form, if this number is different from 1.

Following this, the semantic function I used for the rule is displayed. The format is as described in Chapter 5. In addition to simple set-theoretical functions, the special functions QUANTIF and MEASURE are used with their special definitions assumed as given in Chapter 5. Several other functions are also defined as needed.

After lexical disambiguation by the probabilistic method, there were 1,060 terminal forms, representing 7,046 utterance tokens in ERICA.

1. ADJECTIVE PHRASE RULES

(1,1) adjp -> adj

Types = 199 Tokens = 539
Times used = 214 Times used * Frequency = 556

Semantics: [adj]

An adjp, to characterize it informally, is a string of common adjectives (adj) preceded by an optional adverbial phrase.

Rule (1,1) is the simplest of the rules that introduce such strings.

(1,2) adjp -> adjp adj

Types = 39 Tokens = 63
Times used = 58 Times used * Frequency = 88

Semantics: [adjp] ∩ [adj]

This is the recursive adjective phrase rule. The forms using it are listed in Chapter 4, so I do not repeat them here.

(1,3) adjp -> advp adjp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
7	1	persp link adv adj	
5	1	adv adj	
4	1	pron link adv adj	
3	1	adv adj n	
3	1	persp link neg adv adj	
2	1	link adv adj	
2	2	persp link adv adv adj	3
1	2	adv adv adj n	3
1	5	adv adv adj adj	8
1	1	adv adj n prep pronadj n	
1	1	conj pronadj adv adj	
1	1	conj pron link adv adj	
1	1	conj persp link adv adj	
1	1	int adv adj	
1	1	n link adv adj n	
1	1	neg adv adj	
1	1	persp v adv adj n	
1	1	persp link adv adj n	
1	1	persp link neg art adv adj n	
1	2	persp link adv adv adj pron n	3
1	1	pron link neg adv adj	
1	2	pron link adv adv adj	3

Types = 22 Tokens = 41
Times used = 37 Times used * Frequency = 58

Semantics: MEASURE(<adjp,ADVP>, [advp] , [adjp])

This rule modifies adjective phrases with adverbial phrases. Only one form has two (or more) adjectives together:

1 adv adv adj adj

The original utterance is

1 in here any more

which contains the adverbial phrases 'in#here' and 'any#more', which should be reclassified in the dictionary.

The form

1 adv adv adj n

represents the sentence

very very angry now .

The word 'now' is very likely misclassed in the dictionary.

When two adverbs modify an adjective phrase, there are two semantic interpretations possible, as shown by the following denotations for 'adv adv adj n':

1) MEASURE(<ADJP,ADVP>, [ADV],
MEASURE(<ADJP,ADVP>, [ADV], [ADJ])) ∩ [N]

This first interpretion is that both adverbs modify the adjective in turn.

2) MEASURE(<adjp,ADVP>,
MEASURE(<adjp,ADVP>, [ADV],[ADV]), [ADJ])
∩ [N]

This second interpretation is that the first adverb modifies the second.

Let me elaborate a bit on this ambiguity. The intuition behind the function MEASURE is that the adverb assumes an ordering on the modified set and then extracts a section from that ordering. The other notion of adverbs that I considered in Chapter 5, and rejected, is that the adverb selects a subset of the modified set. (This second more general interpretation seems too non-specific to be helpful in describing the semantics of ERICA.)

No good examples of this ambiguity appear in ERICA to my knowledge. Some fictitious examples are the adjective phrases:

a) somewhat overly protective
b) fairly well considered

For a) the correct order of modification is given by 1), whereas for b) the correct order is 2). Notice that we would, intuitively, group 'overly protective' together, then modify by 'somewhat' in a), whereas in b) the tendency is to group 'fairly well' together.

Of course, some ways of handling the function MEASURE could yield semantic equivalence, but I think that in the above example it is sufficiently clear to indicate that this is not always the case.

The interpretation favored by the probabilistic grammar is 2). The conditional probabilities for the interpretations are:

1) .39

2) .61

All utterances in ERICA that have an adverbial phrase of two or more adverbs, thereafter modifying an adjective phrase, present this semantic ambiguity. The original utterances, listed by the terminal forms involved, follow. (The line beginning '(From: ' indicates the lexical form involved. Often, since lexical disambiguation has occurred, some consolidation has occurred. See Chapter 4. Text beginning with '(REMARK' contains a comment about the previous group of utterances.

(From: persp link adv adv adj)

i was very very scared.
it's very very sharp.

(From: adv adv adj n)
very very angry now.

(From: adv adv adj adj)
in here any more.

(From: persp link adv adv adj pron n)
i be very very careful this morning.

(Remark: 'this morning' is not a predicate nominative as the grammar says it is. Again, this is an adverbial phrase that needs to be reclassified in the dictionary.)

(From: pron link adv adv adj)
those are very very high.

Looking at these utterances involving two adverbs, it is not clear which interpretation is to be favored. If we believe the probabilistic grammar, we would try to analyze 'very very' as an adverbial function, since this interpretation is favored with a conditional probability of .61. One would like to see a greater variety of adverbs to make any claim, since 'very' is the only adverb using this construction in ERICA. See Section II for further discussion of ambiguity.

2. ADVERBIAL PHRASE RULES

(14.1) advp -> adv

Types = 55   Tokens = 260
Times used = 70   Times used * Frequency = 277

Semantics: [adv]

(14.2) advp -> adv advp

Types = 8   Tokens = 29
Times used = 10   Times used * Frequency = 31

Semantics: MEASURE(<ADVP,ADV>, [advp] ,[adv])

Rule (14,2) is the recursive adverbial phrase rule. The forms are given in Chapter 4.

3. QUANTIFIER-ARTICLE RULES

The symbol 'quart' introduces quantifiers and articles into utterances.

Notice that the class of 'qu' contains the cardinal numbers, and the function QUANTIF handles the semantics for these. A more syntactically elegant but semantically equivalent approach would use an added symbol 'card' for the cardinal numbers, making the semantic difference explicit in the syntax. This is to be preferred from a conceptual point of view, since it makes a semantic distinction clear in the syntax. The chief reason that I did not do this is that there appeared to be little difference in the way the various quantifiers were distributed statistically in the corpus and hence no syntactic justification for the added symbol.

This may be a case of the syntax diverging a bit from the semantics. I think that the ERICA corpus offers too little developmental evidence to be certain. We would want to look over a slightly longer period of time. (Erica was between 31 and 33 months old at the time of the recordings.)

The semantics for rules (21,1) and (21,2) is simply the identity function. This is because the function QUANTIF, as described in Chapter 5, is called by the rules that actually introduce the 'quart' into utterances. See rules (22,2), (22,3), (17,4), and (17,5).

(21.1) quart -> qu

Types = 117   Tokens = 277
Times used = 140   Times used * Frequency = 307

Semantics: [qu]

(21.2) quart -> art

Types = 257   Tokens = 821
Times used = 302   Times used * Frequency = 882

Semantics: [art]

4. ADJECTIVE PHRASE RULES -- POSSESSIVE ADJECTIVES

The symbol 'adp' introduces the symbol 'det' to precede strings of common adjectives (adjp). The symbol 'det' then is replaced by either 'pronadj' (pronominal adjectives) or 'padj' (possessive adjectives). These rules are not included among the adjp-rules since, as a probabilistic grammar, GE1 accounts for the fact that possessives usually precede common adjectives. For example, notice the two utterances representing the form

(From: adv link pronadj adj n)

1 here is my big quilt.
1 there is my new <n>.

(Remark: the symbol '<n>' stands for unidentifiable noun.)

I have not found in ERICA a single example of a possessive occurring after a common noun in a modifying phrase. GE1 accounts for this; the price paid is the use of rules that have no apparent semantic content.

(9.1) adp -> adjp

Types = 62 Tokens = 157
Times used = 68 Times used * Frequency = 164

Semantics: [adjp]

(9.2) adp -> det

Types = 115 Tokens = 297
Times used = 139 Times used * Frequency = 327

Semantics: [det]

(9.3) adp -> det adjp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
2	1	adv link pronadj adj n	
2	1	intadv aux pronadj adj n	
2	1	persp v pronadj adj n	
1	1	mod persp v pronadj adj n	
1	1	neg pron link pronadj adj n	

```
1        1        persp link pronadj adj n
1        1        persp mod neg v pronadj adj n
1        1        persp v prep art n adj n prep pronadj adj n
1        1        pn v persp pronadj adj n
1        1        pron link pronadj adj n
1        1        pronadj adj n
1        1        pronadj adj n conj art n
1        1        v pronadj adj n
Types = 13    Tokens = 16
Times used = 13   Times used * Frequency = 16
```

Semantics: [det] $\cap$ [adjp]

5. RULES FOR ADJECTIVE-PHRASES NOT PRECEDING NOUN PHRASES

Several rules introduce adjective phrases that do not precede a noun phrase. These rules are: (7,5), (4,9), (4,12), and (4,41). When an adjective phrase stands alone, the effect of a 'quart' (quantifier or article) must be made on the adjective phrase alone. As an example, consider the form

```
7       persp link qu adj
```

representing

```
4       he's all black.
        he's all green.
        it's all better.
        he is all better.
```

The denotation for these is

if [persp] $\subseteq$ QUANTIF([qu],[adj]) then

TRUE else FALSE.

As I mentioned in Chapter 5, this use of 'all' is not really as a quantifier, but rather an adjective (possibly attributive). Since the uses of 'all' that have this sense are connected with adjective phrases not preceding a noun, the semantics could be modified to handle it easily enough; for example,

```
        if [persp] ⊆ ALL([adjp]) then
              TRUE else FALSE ,
```

using a function ALL to compute the appropriate subset of [adjp]. I am not clear about all the implications that this sort of thing would have.

The qadp-rules generate adjective phrases that do not precede nouns.

<u>(22.1) qadp -> adjp</u>

Types = 43   Tokens = 213
Times used = 49   Times used * Frequency = 220

Semantics: [adjp]

<u>(22.2) qadp -> quart adjp</u>

TERMINAL FORMS

```
Types   No. of            Form    Times rule used on form
      Derivations                 (If different from 1)
----------------------------------------------------------
```

7	1	persp link qu adj	
6	1	qu adj	
3	1	art adj	
2	1	persp link neg qu adj	
1	1	art adj adj	
1	1	persp link art adj	
1	1	persp link neg qu adj adj adj	
1	1	pron link art adj	
1	1	pron link art adj adj adj	

Types = 9 Tokens = 23
Times used = 9 Times used * Frequency = 23

Semantics: QUANTIF([quart] , [adjp])

(22.3) qadp -> quart

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
3	1	pron link art	
2	1	art	
1	1	link qu	
1	1	link pron qu	

Types = 4 Tokens = 7
Times used = 4 Times used * Frequency = 7

Semantics: QUARTC([quart])

The QUARTC function is given by

QUARTC([quart]) = QUANTIF([quart],IMMED)

I list below, by terminal form, the utterances using this function.

3 pron link art

that's a ...
there's a ...
this is a ...

(Remark: These appear to be fragments.)

2 art
2 a.

1 link qu
... is this.

(Remark: Lexical disambiguation appears to have failed on 'link qu', since the word 'this' is probably a pronoun rather than a quantifier. It is, of course, classed in the dictionary as both.)

1 link pron qu
is another one?

(Remark: Another failure of lexical disambiguation.)

Most of these utterances appear to be fragmentary, so there is little to conclude about the value of the QUARTC function.

(22.4) qadp -> det

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
10	1	pronadj	
7	1	pron link padj	
6	1	pron link pronadj	
4	1	padj	
2	1	persp link padj	
1	1	neg pron link pronadj	
1	1	persp link pronadj	

Types = 7   Tokens = 31
Times used = 7   Times used * Frequency = 31

Semantics: [det]

Notice in the above forms for (22,4) that the symbol 'det' does not occur in any form; this is because it is, of course, a non-terminal symbol of the grammar GE1. 'det' introduces possessive adjectives ('padj') and pronominal adjectives ('pronadj') into utterances through rules (10,1) and (10,2).

(22,5) qadp -> det adjp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
1	1	conj pronadj adv adj	
1	1	pronadj adj	

Types = 2   Tokens = 2
Times used = 2   Times used * Frequency = 2

Semantics: [det] ∩ [adjp]

6. RULES INTRODUCING POSSESSIVES

The symbols 'padj' and 'pronadj' are the possessive adjectives, which are introduced through the 'det' symbol.

(10,1) det -> pronadj

Types = 121 Tokens = 312
Times used = 144 Times used * Frequency = 341

Semantics: [ronadj]

(10,2) det -> padj

Types = 16 Tokens = 34
Times used = 17 Times used * Frequency = 35

Semantics: [padj]

7. NOUN-PHRASE RULES

Several sets of rules introduce noun phrases. The proliferation of symbols is, again, to make GE1 a reasonable probabilistic grammar. This proliferation is *prima facie* disturbing, especially since many of the rules have little semantic content. The explanation is that noun-phrase constructions appear rather differently when used in different parts of the utterance. In particular, noun-phrases that stand as the whole utterance are rather unlike noun-phrases that serve as the objects of prepositions. See Chapter 4 for the parameters associated with the rules of GE1.

(2,1) nounp -> pn

Types = 112 Tokens = 234
Times used = 137 Times used * Frequency = 269

Semantics: [pn]

(2.2) nounp -> n

Types = 650 Tokens = 2590
Times used = 1030 Times used * Frequency = 3269

Semantics: [n]

(2.3) nounp -> pron

Types = 295 Tokens = 1239
Times used = 385 Times used * Frequency = 1421

Semantics: [pron]

(13.1) np -> npsub prepp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
14	2	persp v pron prep pron	
5	2	persp v persp prep art n	
4	2	mod persp v pron prep pron	
4	2	v persp prep art n	
3	2	persp mod v pron prep pron	
3	2	persp mod v persp prep persp	
2	2	persp v n prep persp	
2	2	persp v persp prep pn	
2	2	persp v pron prep art n	
2	2	persp v art n prep persp	
2	2	persp v persp prep persp	
2	2	persp v n prep pronadj n	
2	2	persp v adj n prep persp	
2	2	persp v pron prep pronadj n	
2	2	persp mod v art n prep persp	

2	2	persp v persp prep pronadj n
2	2	persp aux v n prep pronadj n
2	2	persp mod neg v art n prep pronadj n
2	1	pron link qu n prep persp
2	2	v persp prep n
2	1	v persp pron prep pron
2	2	v pron prep persp
1	1	aff prep n prep persp
1	2	aux n prep art n
1	2	aux pron prep art n
1	2	conj persp v pron prep pron
1	2	conj art n prep persp v art n
1	2	conj mod qu n prep n v neg persp
1	2	int persp v pron prep pron
1	2	intadv aux art n prep art n
1	2	inter pron prep art n
1	2	inter link pron prep pronadj n
1	2	mod persp v pron prep n
1	1	mod persp v persp prep n n
1	1	mod persp v n prep art n n
1	2	mod persp v pron prep art n
1	2	mod persp v persp prep qu n
1	2	mod persp v pronadj n prep n
1	2	mod persp v persp prep art n
1	2	mod persp v pronadj n prep pron
1	1	mod persp v pron prep pron art n
1	1	mod persp v prep pronadj n n prep art n
1	1	n n v n n prep art n
1	1	n n v prep pronadj n prep persp
1	2	n pn aux v n prep art n
1	2	n v art n prep persp
1	2	n v pron prep persp
1	2	n v persp prep persp
1	2	n v pronadj n prep art n
1	2	n v pronadj n prep persp
1	2	neg persp mod neg v pron prep pron
1	2	persp v art n prep n
1	2	persp v n prep art n
1	2	persp mod v pron prep n
1	2	persp v art n prep pron
1	1	persp link n prep art n
1	2	persp v qu n prep persp
1	1	persp v prep n prep pron
1	2	persp mod neg v n prep n
1	2	persp aux v n prep persp
1	2	persp v pron prep padj n
1	1	persp v prep persp prep n
1	1	persp aux v prep n prep n
1	1	persp v pron prep art pn n
1	2	persp v pron prep qu n aux

1	2	persp mod v persp prep pron	
1	2	persp mod v persp prep art n	
1	2	persp v art n prep art adj n	
1	1	persp v pron prep art pron n	
1	2	persp mod neg v n prep persp	
1	1	persp v persp pron prep persp	
1	1	persp v pron prep pronadj n n	
1	1	persp mod v persp prep persp n	
1	2	persp v art adj pron prep pron	
1	2	persp aux v qu pron prep persp	
1	1	persp mod v prep pron prep pron	
1	2	persp mod v pronadj n prep persp	
1	2	persp mod v persp prep pronadj n	
1	2	persp mod neg v adj n prep persp	
1	1	persp mod v persp prep pronadj n n	
1	1	persp v prep art n adj n prep pronadj adj n	
1	2	pn n mod neg v pron prep pron	
1	1	pn v art n prep n n	
1	2	pn v pron prep persp	
1	1	pron link n prep art n	
1	1	pron link n prep persp	
1	1	pron link pron prep pron	
1	2	pronadj n v art n prep persp	
1	2	v art n prep pron	
1	1	v art pron pron prep persp	
1	2	v n prep n	
1	2	v n prep persp	
1	2	v persp prep persp	
1	3	v persp prep n prep art n n	4
1	2	v pron prep art n n	
1	2	v pronadj n prep pronadj n	
1	2	v qu n prep art n	

Types = 97 Tokens = 140

Times used = 100 Times used * Frequency = 143

Semantics: [npsub] ∩ [prepp]

This rule lets a prepositional phrase modify a noun phrase. I have included the complete list of forms here to supplement the discussion of semantic ambiguity in Section II below.

Notice that many of these forms have two derivations. The reason for this grammatical ambiguity is that the prepositional phrase may alternatively be viewed as an object of the verb instead of as a modifier to the noun-phrase.

(NOTE: Here the reader may note that rule (13,2) has been removed from the grammar. I have retained this numbering so that I don't confuse the computer program that formats all the tables of this work.)

(13,3) np -> npsub conj npsub

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
2	1	pron link n conj n	
1	1	adj adj n conj pron aux v art n	
1	1	art n conj art n v prep art n	
1	1	conj pn conj pn aux v prep n	
1	1	n n conj persp v pron	
1	1	persp v pn conj pn	
1	1	persp link art n conj n	
1	1	persp v n conj pronadj n	
1	1	persp v pronadj pn conj n	
1	1	persp v pron conj qu pron	
1	1	persp v prep art n conj art n	
1	1	persp conj persp mod v qu pron	
1	1	persp v art adj pron conj art n	
1	1	persp mod neg v art n n conj art n	
1	1	persp mod neg v pronadj n conj pronadj n	
1	1	pn conj pn mod neg v art n	
1	1	pn prep pn conj persp	
1	1	prep pn conj pn	
1	1	prep pronadj n conj n	
1	1	pron link pn conj pn	
1	1	pron link pn pn conj pn	
1	1	pron link qu n conj art n	
1	1	pron link art n conj art n	
1	1	pronadj n conj pronadj n prep persp v	

Types = 24   Tokens = 25

Times used = 24   Times used * Frequency = 25

Semantics: $([npsub]) \cup ([npsub])$

This rule conjoins noun phrases together with conjunctions. I believe the correct function is union, as in

2. pron link n conj n

representing

that's mommy and daddy.
there's mommy and daddy.

Consider

pronadj n conj pronadj n prep persp v

which has the denotation:

$$\text{if } (([pronadj] \cap [n]) \cup ([pronadj] \cap [n])) \cap \{a \mid (\exists \langle a,b \rangle \in [prep])(b \in [persp])\} \subseteq [v] \text{ then TRUE else FALSE}.$$

The original utterance is

my mommy and daddy 'fore it rain.

It contains the phrase 'fore it rain' as an adverbial expression. Hence, the analysis is incorrect in this case, and this is the only utterance represented by the form.

The use of the union function seems appropriate for most of the utterances requiring rule (13.3). The 'conj' is almost always the word 'and'.

(13.4) np -> npsub

Types = 868 Tokens = 3518
Times used = 1751 Times used * Frequency = 5624

Semantics: [npsub]

(17.1) npsub -> persp

Types = 525 Tokens = 2291
Times used = 692 Times used * Frequency = 2744

Semantics: [persp]

(17.2) npsub -> nounp

Types = 559 Tokens = 2546
Times used = 903 Times used * Frequency = 3293

Semantics: [nounp]

(17.3) npsub -> adp nounp

Types = 188 Tokens = 468
Times used = 220 Times used * Frequency = 507

Semantics: [adp] ∩ [nounp]

(17.4) npsub -> quart nounp

Types = 288 Tokens = 937
Times used = 356 Times used * Frequency = 1026

Semantics: QUANTIF([quart] , [nounp])

Rules (17,4) and (17,5) generate noun-phrases modified by a 'quart'.

(17,5) npsub -> quart adjp nounp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
12	1	persp v art adj n	
10	1	pron link art adj n	
9	1	persp v art adj pron	
8	1	art adj n	
6	1	persp link art adj n	
6	1	pron link art adj pron	
3	1	conj art adj n	
3	1	qu adj n	
3	1	qu adj n n	
2	1	neg pron link art adj n	
2	1	persp v qu adj n	
2	1	persp link art adj pron	
2	1	persp link neg art adj n	
2	1	persp link art adj adj n	
2	1	pron link art adj adj n	
2	1	v art adj n	
1	1	adv link art adj adj pron	
1	1	art adj n n	
1	1	art adj pron	
1	1	art adj adj n	
1	1	art adj adj n v	
1	1	art adj adj pron	
1	1	art adj adj adj n	
1	1	art adj pron persp v	
1	1	conj art adj adj n	
1	1	conj persp v art adj n	
1	1	conj adv link art adj n	
1	1	conj pron link art adj pn	
1	1	conj persp link art adj n n	
1	1	conj persp v art adj adj pron	
1	1	conj pron link art adj adj pron	
1	1	int pron aux v art adj n	
1	1	int pron link art adj adj n	
1	1	intadv aux qu adj n	
1	1	intadv aux art adj n	

```
1       1         intadv persp v art adj n
1       1         inter link qu adj n
1       1         mod persp v qu adj n
1       1         mod persp v art adj pron
1       1         n link art adj n
1       1         n link art adj adj n
1       1         n mod v prep art adj n
1       1         neg persp link art adj n
1       1         persp art adj adj n
1       1         persp v art adj n n
1       1         persp mod v qu adj n
1       1         persp mod v art adj n
1       1         persp v art adj adj n
1       1         persp mod v art adj pron
1       1         persp mod neg v qu adj n
1       1         persp v prep art adj pron
1       1         persp link prep art adj n
1       1         persp mod neg v art adj n n
1       2         persp v art n prep art adj n     2
1       1         persp link neg art adv adj n
1       2         persp v art adj pron prep pron   2
1       1         persp v art adj pron conj art n
1       1         pn link art adj adj n
1       1         prep art adj n
1       1         pron art adj n
1       1         pron art adj adj n
1       1         pron link neg art adj n
1       1         pron link pron qu adj n
1       1         pron link art adj pron art n
1       1         qu adj adj n
1       1         qu adj n v n n
1       1         qu adj n mod neg
1       1         qu pron link art adj n
1       1         v art adj pron
1       1         v persp art adj pron
1       1         v qu adj n
Types = 71    Tokens = 129
Times used = 73   Times used * Frequency = 131
```

Semantics: QUANTIF([quart] ,([adjp] ∩ [nounp]))

8. VERB-PHRASE RULES

(5.1) vbl -> auxilp vp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
402	1	persp mod neg v	
33	1	persp aux v	
29	1	persp mod v persp	
18	1	persp mod v	
16	1	persp mod v pron	
14	1	persp mod neg v persp	
13	1	persp mod neg v n	
11	1	persp aux v n	
10	1	persp mod v persp prep	
9	1	persp mod neg v pron	
7	1	persp aux v pron	
7	1	persp aux v art n	
7	1	persp mod v prep persp	
7	1	persp aux v prep art n	
6	1	persp aux neg v	
6	1	persp mod v art n	
6	1	persp aux v pronadj n	
6	1	persp mod neg v pronadj n	
6	1	persp mod neg v persp prep	
5	1	inter aux v prep	
5	1	persp aux v prep	
5	1	persp aux v persp	
4	1	persp mod v n	
4	1	persp mod v prep	
4	1	persp mod neg v art n	
4	1	persp aux v prep persp	
4	1	qu n aux v	
3	1	art n aux v prep	
3	1	conj persp aux v	
3	1	n aux v	
3	1	n persp mod v persp	
3	1	persp mod v qu n	
3	1	persp aux v prep n	
3	1	persp mod neg v prep	
3	1	persp mod neg v qu n	
3	1	persp mod v pron prep	
3	1	persp mod v prep pron	
3	1	persp aux v prep pronadj n	
3	2	persp mod v pron prep pron	2
3	2	persp mod v persp prep persp	2
3	1	persp mod neg v pronadj n prep	
2	1	art n aux v	
2	1	art n persp mod v	
2	1	conj persp mod neg v prep	

2	1	neg persp mod neg v prep	
2	1	persp aux v pn	
2	1	persp aux v qu n	
2	1	persp aux neg v prep	
2	1	persp aux neg v pron	
2	1	persp mod v pronadj n	
2	1	persp aux neg v persp	
2	1	persp aux v prep qu n	
2	1	persp mod v persp art n	
2	1	persp aux v persp art n	
2	1	persp aux v prep art n n	
2	1	persp mod v prep pronadj n	
2	1	persp mod neg v art n prep	
2	2	persp mod v art n prep persp	2
2	2	persp aux v n prep pronadj n	2
2	1	persp mod neg v prep pronadj n	
2	2	persp mod neg v art n prep pronadj n	2
2	1	pron aux v	
2	1	pron mod v prep	
2	1	pron aux v prep	
1	1	adj adj n mod neg v qu n	
1	1	adj adj n conj pron aux v art n	
1	1	adj n mod negzv	
1	1	adj n persp mod neg v art n	
1	1	adv persp aux v	
1	1	aff persp mod v persp	
1	1	art n mod neg v	
1	1	art n mod neg v neg	
1	1	art n mod neg v prep	
1	1	conj art n mod v	
1	1	conj pn n aux v n	
1	1	conj persp mod v n	
1	1	conj persp aux v n	
1	1	conj persp mod v pron	
1	1	conj persp mod neg v n	
1	1	conj pn mod v prep persp	
1	1	conj persp mod neg v neg	
1	1	conj pron mod neg v prep	
1	1	conj persp aux v prep pron	
1	1	conj persp mod v prep art n	
1	1	conj persp aux v prep adj n	
1	1	conj persp mod v art n prep	
1	1	conj pn conj pn aux v prep n	
1	1	conj persp mod neg v persp prep	
1	1	conj persp persp mod v prep pronadj n	
1	1	int persp aux v	
1	1	int pron aux v art adj n	
1	1	int persp mod v persp prep	
1	1	inter persp modzv	
1	1	mod neg v pronadj n prep	

1	1	n aux v neg persp	
1	1	n mod v	
1	1	n mod neg v	
1	1	n mod v pron	
1	1	n mod v art n	
1	1	n mod neg v n	
1	1	n mod neg v prep	
1	1	n mod neg v art n	
1	1	n mod v persp prep	
1	1	n mod v prep art adj n	
1	1	n n mod v prep pron	
1	1	n n mod neg v art n	
1	1	n persp mod v	
1	1	n persp mod neg v	
1	1	n persp mod v prep	
1	1	n persp mod v persp prep	
1	1	n persp aux v prep art n	
1	1	n pn aux v prep	
1	1	n pn mod neg v n	
1	2	n pn aux v n prep art n	2
1	1	n pn mod neg v prep art n	
1	1	neg pronadj n aux v	
1	1	neg persp aux neg v	
1	1	neg persp mod v persp	
1	1	neg persp mod neg v pron	
1	1	neg persp aux v prep persp	
1	1	neg persp mod v prep pronadj n	
1	2	neg persp mod neg v pron prep pron	2
1	1	persp mod v n n	
1	1	persp aux v n n	
1	1	persp aux v int	
1	1	persp mod v adj n	
1	1	persp aux v adj n	
1	1	persp mod v prep n	
1	1	persp mod v art n n	
1	1	persp mod v persp n	
1	1	persp mod v prep pn	
1	1	persp mod neg v n n	
1	1	persp mod neg v int	
1	1	persp aux v art n n	
1	1	persp mod v qu pron	
1	1	persp mod v adj pron	
1	1	persp mod v qu adj n	
1	1	persp mod v art adj n	
1	1	persp mod neg v n aff	
1	1	persp mod neg v adj n	
1	1	persp aux v prep pron	
1	1	persp mod v prep qu n	
1	1	persp mod v art n prep	
1	1	persp mod neg v n prep	

1	1	persp mod neg v prep n	
1	1	persp mod neg v art n n	
1	2	persp mod v pron prep n	2
1	1	persp mod v prep padj n	
1	1	persp mod v art adj pron	
1	2	persp mod neg v n prep n	2
1	2	persp aux v n prep persp	2
1	1	persp mod neg v qu adj n	
1	1	persp mod neg v prep pron	
1	1	persp mod v prep adj pron	
1	1	persp mod v qu pron art n	
1	1	persp aux v prep n prep n	
1	1	persp mod v pronadj n prep	
1	1	persp mod neg v prep persp	
1	1	persp mod neg v prep art n	
1	1	persp aux neg v prep art n	
1	1	persp mod neg v art adj n n	
1	2	persp mod v persp prep pron	2
1	2	persp mod v persp prep art n	2
1	2	persp mod neg v n prep persp	2
1	1	persp mod neg v pronadj adj n	
1	1	persp mod neg v art pron pron	
1	1	persp aux v prep persp padj n	
1	1	persp mod v persp prep persp n	
1	1	persp conj persp mod v qu pron	
1	2	persp aux v qu pron prep persp	2
1	1	persp mod v prep pron prep pron	
1	1	persp mod neg v prep art pron n	
1	2	persp mod v pronadj n prep persp	2
1	2	persp mod v persp prep pronadj n	2
1	2	persp mod neg v adj n prep persp	2
1	1	persp mod v persp prep pronadj n n	
1	1	persp mod neg v art n n conj art n	
1	1	persp mod neg v pronadj n conj pronadj n	
1	1	pn aux v persp	
1	1	pn conj pn mod neg v art n	
1	1	pn mod neg v persp	
1	1	pn mod v prep persp	
1	2	pn n mod neg v pron prep pron	2
1	1	pron aux v n	
1	1	pron aux v pn	
1	1	pron mod v pron	
1	1	pron aux v pron	
1	1	pron mod v persp	
1	1	pron aux v art n	
1	1	pron aux v persp	
1	1	pron mod neg v n	
1	1	pron mod neg v prep	
1	1	pron mod v pronadj n	
1	1	pron mod v prep persp	

```
1       1       pron persp aux v prep
1       1       pronadj n aux v pronadj n
1       1       qu n mod neg v pron
1       1       qu n mod v prep pronadj n
Types = 198    Tokens = 870
Times used = 216          Times used * Frequency = 895
```

Semantics: AUXFCN([auxilp],[vp])

AUXFCN is defined by the following:

AUXFCN([auxilp],[vp]) =

IF auxilp does not contain (syntactically) any member of the class 'neg',
THEN $[auxilp] \cap [vp]$
ELSE
$(D^3 \cup D^2 \cup D) \sim ([auxilp] \cap [vp])$,

where D is the domain of the model $\mathfrak{A}$.

Notice below the semantics of rule (16,2), which effectively ignores the 'neg' in the denotation [auxilp].

From the view of semantics, some of the rules that introduce the negating particle ('neg') are awkward. (Rules discussed here are (16,2) and (19,2).) These rules introduce 'neg' at a point in the sentence where the complementation function cannot be used on the set to be complemented, since it is not available at that point in the generation. Instead, the effect of complementation is handled later by the special function AUXFCN.

Syntactically, however, these rules describe the generation of the utterances in question very well. Allowing for the generation of 'neg' at the right level of the utterance would necessitate a proliferation of rules. I stress that this is no problem in the semantics, only a slight slippage between the surface syntax and the semantics. I have chosen to proliferate rules of the grammar only when it was either necessary from a semantic view, or to improve the probabilistic fit. Adding rules to introduce 'neg' at the elegant point would not have been justified in either of these ways.

(5.2) vbl -> vp

Types = 256 Tokens = 897
Times used = 284 Times used * Frequency = 951

Semantics: [vp]

(16.1) auxilp -> auxil

Types = 243 Tokens = 703
Times used = 267 Times used * Frequency = 736

Semantics: [auxil]

(16.2) auxilp -> auxil neg

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)

402	1	persp mod neg v	
23	1	mod neg v	
18	1	persp mod neg	
14	1	persp mod neg v persp	
13	1	persp mod neg v n	
10	1	mod neg persp	
9	1	persp mod neg v pron	
8	1	neg persp aux neg	
8	1	persp aux neg	
6	1	aux neg persp	
6	1	persp aux neg v	
6	1	persp mod neg v pronadj n	
6	1	persp mod neg v persp prep	
4	1	neg persp mod neg	
4	1	persp mod neg v art n	
3	1	n mod neg	
3	1	persp mod neg v prep	
3	1	persp mod neg v qu n	
3	1	persp mod neg v pronadj n prep	
3	1	pron aux neg	
2	1	aux neg pron	
2	1	conj persp mod neg v prep	
2	1	neg persp mod neg v prep	
2	1	persp aux neg v prep	
2	1	persp aux neg v pron	
2	1	persp aux neg v persp	
2	1	persp mod neg v art n prep	
2	1	persp mod neg v prep pronadj n	
2	2	persp mod neg v art n prep pronadj n	2
2	1	pron mod neg	
1	1	adj adj n mod neg v qu n	
1	1	adj n mod neg v	
1	1	adj n persp mod neg v art n	
1	1	art n mod neg	
1	1	art n mod neg v	
1	1	art n mod neg v neg	
1	1	art n mod neg v prep	
1	1	conj persp mod neg v n	
1	1	conj persp mod neg v neg	
1	1	conj pron mod neg v prep	
1	1	conj persp mod neg v persp prep	
1	1	int mod neg qu n v prep	
1	1	intadv mod neg persp v n	
1	1	mod neg v int	
1	1	mod neg n pron	
1	1	mod neg persp n	
1	1	mod neg v pronadj n prep	
1	1	n mod neg v	
1	1	n mod neg v n	
1	1	n mod neg v prep	

```
1       1       n mod neg v art n
1       1       n n mod neg v art n
1       1       n persp mod neg v
1       1       n pn mod neg v n
1       1       n pn mod neg v prep art n
1       1       neg n mod neg
1       1       neg persp aux neg v
1       1       neg persp mod neg v pron
1       2       neg persp mod neg v pron prep pron     2
1       1       persp mod neg v n n
1       1       persp mod neg v int
1       1       persp mod neg v n aff
1       1       persp mod neg v adj n
1       1       persp mod neg v n prep
1       1       persp mod neg v prep n
1       1       persp mod neg v art n n
1       2       persp mod neg v n prep n               2
1       1       persp mod neg v qu adj n
1       1       persp mod neg v prep pron
1       1       persp mod neg v prep persp
1       1       persp mod neg v prep art n
1       1       persp aux neg v prep art n
1       1       persp mod neg v art adj n n
1       2       persp mod neg v n prep persp           2
1       1       persp mod neg v pronadj adj n
1       1       persp mod neg v art pron pron
1       1       persp mod neg v prep art pron n
1       2       persp mod neg v adj n prep persp       2
1       1       persp mod neg v art n n conj art n
1       1       persp mod neg v pronadj n conj pronadj n
1       1       pn conj pn mod neg v art n
1       1       pn mod neg
1       1       pn mod neg v persp
1       2       pn n mod neg v pron prep pron          2
1       1       pron mod neg v n
1       1       pron mod neg v prep
1       1       pronadj n aux neg
1       1       qu adj n mod neg
1       1       qu n mod neg v pron
Types = 89    Tokens = 631
Times used = 95   Times used * Frequency = 638
```

Semantics: [auxil]

Notice that the semantics for this does not include any effect of the negating particle. See the discussion following rule (5,1) for an explanation, and also Section II.

(15.1) auxil -> aux

Types = 112 Tokens = 328
Times used = 120 Times used * Frequency = 337

Semantics: [aux]

(15.2) auxil -> mod

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
402	1	persp mod neg v	
29	1	persp mod v persp	
27	1	mod persp v persp	
24	1	adv persp mod	
23	1	mod neg v	
19	1	persp mod	
18	1	persp mod v	
18	1	persp mod neg	
17	1	mod persp v pron	
16	1	persp mod v pron	
14	1	persp mod neg v persp	
13	1	mod persp v prep persp	
13	1	persp mod neg v n	
10	1	mod neg persp	
10	1	mod persp v	
10	1	persp mod v persp prep	
9	1	aff persp mod	
9	1	mod persp v art n	
9	1	persp mod neg v pron	
8	1	mod v	
7	1	persp mod v prep persp	
6	1	persp v neg mod	
6	1	persp mod v art n	
6	1	persp mod neg v pronadj n	

6	1	persp mod neg v persp prep	
5	2	mod persp	
5	1	mod persp v n	
5	1	mod persp v prep	
5	1	mod persp v prep n	
4	1	mod persp v pronadj n	
4	2	mod persp v pron prep pron	2
4	1	neg mod persp v persp	
4	1	neg persp mod neg	
4	1	persp mod v n	
4	1	persp mod v prep	
4	1	persp mod neg v art n	
3	1	mod persp v qu n	
3	1	mod persp v prep pron	
3	1	mod persp v prep pronadj n	
3	1	n mod neg	
3	1	n persp mod v persp	
3	1	persp mod v qu n	
3	1	persp mod neg v prep	
3	1	persp mod neg v qu n	
3	1	persp mod v pron prep	
3	1	persp mod v prep pron	
3	2	persp mod v pron prep pron	2
3	2	persp mod v persp prep persp	2
3	1	persp mod neg v pronadj n prep	
2	1	art n persp mod v	
2	1	conj persp mod neg v prep	
2	1	intadv mod persp v	
2	1	inter mod persp v	
2	2	mod pron	
2	1	mod persp v n n	
2	1	mod persp v adj n	
2	1	mod persp v qu pron	
2	1	mod persp v persp prep	
2	1	neg persp mod neg v prep	
2	1	persp mod v pronadj n	
2	1	persp mod v persp art n	
2	1	persp mod v prep pronadj n	
2	1	persp mod neg v art n prep	
2	2	persp mod v art n prep persp	2
2	1	persp mod neg v prep pronadj n	
2	2	persp mod neg v art n prep pronadj n	2
2	1	pron mod neg	
2	1	pron mod v prep	
1	1	adj adj n mod neg v qu n	
1	1	adj n mod neg v	
1	1	adj n persp mod neg v art n	
1	1	aff mod persp n	
1	1	aff persp mod v persp	
1	1	art n mod neg	

1	1	art n mod neg v	
1	1	art n mod neg v neg	
1	1	art n mod neg v prep	
1	1	conj pron mod	
1	1	conj art n mod v	
1	1	conj persp mod v n	
1	1	conj persp mod v pron	
1	1	conj persp mod neg v n	
1	1	conj pn mod v prep persp	
1	1	conj persp mod neg v neg	
1	1	conj pron mod neg v prep	
1	1	conj persp mod v prep art n	
1	1	conj persp mod v art n prep	
1	1	conj persp mod neg v persp prep	
1	2	conj mod qu n prep n v neg persp	2
1	1	conj persp persp mod v prep pronadj n	
1	1	int mod persp v	
1	1	int mod persp v n	
1	1	int mod neg qu n v prep	
1	1	int persp mod v persp prep	
1	1	intadv mod neg persp v n	
1	1	inter persp mod v	
1	1	mod art n n	
1	1	mod art n v n	
1	1	mod n v n n	
1	1	mod n v persp	
1	1	mod neg v int	
1	1	mod neg n pron	
1	1	mod neg persp n	
1	1	mod n v prep art n	
1	1	mod neg v pronadj n prep	
1	1	mod pn v n	
1	1	mod persp n	
1	2	mod pronadj n	
1	1	mod pron v prep	
1	1	mod pronadj n v	
1	1	mod persp v art n n	
1	1	mod persp v persp n	
1	1	mod pronadj n v pron	
1	1	mod persp v qu adj n	
1	1	mod pron v prep art n	
1	1	mod persp v qu pron n	
1	1	mod persp v prep qu n	
1	1	mod persp v prep pn pn	
1	1	mod persp v prep art n	
1	2	mod persp v pron prep n	2
1	1	mod persp v prep art n n	
1	1	mod persp v art adj pron	
1	1	mod persp v pronadj adj n	
1	1	mod persp v persp prep n n	

1	1	mod persp v n prep art n n	
1	2	mod persp v pron prep art n	2
1	2	mod persp v persp prep qu n	2
1	2	mod persp v pronadj n prep n	2
1	2	mod persp v persp prep art n	2
1	2	mod persp v pronadj n prep pron	2
1	1	mod persp v pron prep pron art n	
1	1	mod persp v prep pronadj n n prep art n	
1	1	n mod v	
1	1	n mod neg v	
1	1	n mod v pron	
1	1	n mod v art n	
1	1	n mod neg v n	
1	1	n mod neg v prep	
1	1	n mod neg v art n	
1	1	n mod v persp prep	
1	1	n mod v prep art adj n	
1	1	n n mod v prep pron	
1	1	n n mod neg v art n	
1	1	n persp mod v	
1	1	n persp mod neg v	
1	1	n persp mod v prep	
1	1	n persp mod v persp prep	
1	1	n pn mod neg v n	
1	1	n pn mod neg v prep art n	
1	2	neg mod persp	
1	1	neg mod persp v pron	
1	1	neg mod persp v prep persp n	
1	1	neg n mod neg	
1	1	neg persp mod v persp	
1	1	neg persp mod neg v pron	
1	1	neg persp mod v prep pronadj n	
1	2	neg persp mod neg v pron prep pron	2
1	1	persp mod v n n	
1	1	persp v persp mod	
1	1	persp mod v adj n	
1	1	persp mod v prep n	
1	1	persp mod v art n n	
1	1	persp mod v persp n	
1	1	persp mod v prep pn	
1	1	persp mod neg v n n	
1	1	persp mod neg v int	
1	1	persp mod v qu pron	
1	1	persp mod v adj pron	
1	1	persp mod v qu adj n	
1	1	persp mod v art adj n	
1	1	persp mod neg v n aff	
1	1	persp mod neg v adj n	
1	1	persp mod v prep qu n	
1	1	persp mod v art n prep	

```
1       1       persp mod neg v n prep
1       1       persp mod neg v prep n
1       1       persp mod neg v art n n
1       2       persp mod v pron prep n                 2
1       1       persp mod v prep padj n
1       1       persp mod v art adj pron
1       2       persp mod neg v n prep n                2
1       1       persp mod neg v qu adj n
1       1       persp mod neg v prep pron
1       1       persp mod v prep adj pron
1       1       persp mod v qu pron art n
1       1       persp mod v pronadj n prep
1       1       persp mod neg v prep persp
1       1       persp mod neg v prep art n
1       1       persp mod neg v art adj n n
1       2       persp mod v persp prep pron             2
1       2       persp mod v persp prep art n            2
1       2       persp mod neg v n prep persp            2
1       1       persp mod neg v pronadj adj n
1       1       persp mod neg v art pron pron
1       1       persp mod v persp prep persp n
1       1       persp conj persp mod v qu pron
1       1       persp mod v prep pron prep pron
1       1       persp mod neg v prep art pron n
1       2       persp mod v pronadj n prep persp        2
1       2       persp mod v persp prep pronadj n        2
1       2       persp mod neg v adj n prep persp        2
1       1       persp mod v persp prep pronadj n n
1       1       persp mod neg v art n n conj art n
1       1       persp mod neg v pronadj n conj pronadj n
1       1       pn conj pn mod neg v art n
1       1       pn mod
1       1       pn mod neg
1       1       pn mod neg v persp
1       1       pn mod v prep persp
1       2       pn n mod neg v pron prep pron           2
1       1       pron mod
1       1       pron mod v pron
1       1       pron mod v persp
1       1       pron mod neg v n
1       1       pron mod neg v prep
1       1       pron mod v pronadj n
1       1       pron mod v prep persp
1       1       qu adj n mod neg
1       1       qu n mod neg v pron
1       1       qu n mod v prep pronadj n
Types = 220    Tokens = 1006
Times used = 242          Times used * Frequency = 1037
```

Semantics: [mod]

(3.1) vp -> verb

Types = 86 Tokens = 778
Times used = 86 Times used * Frequency = 778

Semantics: [verb]

(3.2) vp -> verb prep

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
14	1	persp v prep	
5	1	inter aux v prep	
5	1	mod persp v prep	
5	1	persp aux v prep	
4	1	persp mod v prep	
3	1	art n aux v prep	
3	1	conj persp v prep	
3	1	persp mod neg v prep	
3	1	pron v prep	
2	1	aux persp v prep	
2	1	conj pron v prep	
2	1	conj persp mod neg v prep	
2	1	inter persp v prep	
2	1	n v prep	
2	1	neg persp mod neg v prep	
2	1	persp aux neg v prep	
2	1	pron mod v prep	
2	1	pron aux v prep	
1	1	adj adj n v prep	
1	1	art n v prep	
1	1	art n mod neg v prep	
1	1	conj pron mod neg v prep	
1	1	int mod neg qu n v prep	
1	1	mod pron v prep	
1	1	n mod neg v prep	
1	1	n n v prep	

```
1        1        n persp v prep
1        1        n persp mod v prep
1        1        n pn aux v prep
1        1        pn pn v prep
1        1        pron mod neg v prep
1        1        pron persp aux v prep
1        1        qu n v prep
1        1        v prep pronadj n prep
Types = 34    Tokens = 79
Times used = 34   Times used * Frequency = 79
```

Semantics: [COMBINE([verb] , PREP)]

COMBINE is a purely syntactic function, discussed in Chapter 5. It joins a verb to its associated preposition prior to semantic analysis. This is reasonable enough, as in some of the following utterances represented by

14 persp v prep.

(From: persp v prep,adv)

he comed out.
he stand up.
he wake up.
i get up.
i get in.
they climb up.

(From: persp v,mod prep)

3 i want to
he wants to.

(Remark: Here it is incorrect to COMBINE the verb 'want' with the preposition 'to'.)

(From: persp v prep)

he looking for...
it turn on.
she talking about.

(From: persp v,mod prep,adv)

ne go out.

This rule, (3,2), has a minor problem when used in conjunction with rule (11,2). That difficulty is discussed below.

(3,3) vp -> verb np

Types = 228 Tokens = 768
Times used = 230 Times used * Frequency = 770

Semantics: $\{ a \mid (\exists \langle a,b \rangle \in [verb])(b \in [np]) \}$

(3,4) vp -> verb np np

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
15	1	persp v n n	
4	1	persp v persp n	
2	1	aux persp v qu n n	
2	1	mod persp v n n	
2	1	persp v art n n	
2	1	persp v art pron pron	
2	1	persp mod v persp art n	
2	1	persp aux v persp art n	
1	1	conj persp v qu n n	
1	1	conj persp v pronadj n n	
1	1	mod n v n n	
1	1	mod persp v art n n	
1	1	mod persp v persp n	
1	1	mod persp v qu pron n	
1	1	mod persp v persp prep n n	
1	1	mod persp v n prep art n n	
1	1	mod persp v pron prep pron art n	

```
1        1        n adj n v n pn
1        1        n n v n n prep art n
1        1        neg persp v persp pron
1        1        persp v pn n
1        1        persp v qu n n
1        1        persp mod v n n
1        1        persp v neg n n
1        1        persp v adj n n
1        1        persp aux v n n
1        1        persp v art pron n
1        1        persp v pron art n
1        1        persp v art n pron
1        1        persp mod v art n n
1        1        persp v art n art n
1        1        persp v art adj n n
1        1        persp mod v persp n
1        1        persp v pronadj n n
1        1        persp mod neg v n n
1        1        persp aux v art n n
1        1        persp mod neg v art n n
1        1        persp v pronadj pron pron
1        1        persp mod v qu pron art n
1        1        persp v pron prep art pn n
1        1        persp mod neg v art adj n n
1        1        persp v pron prep art pron n
1        1        persp v persp pron prep persp
1        1        persp mod neg v art pron pron
1        1        persp v pron prep pronadj n n
1        1        persp mod v persp prep persp n
1        1        persp mod v persp prep pronadj n n
1        1        persp mod neg v art n n conj art n
1        1        pn v art n n
1        1        pn v persp pron
1        1        pn v persp adj n
1        1        pn v art n prep n n
1        1        pn v persp pronadj adj n
1        1        pron v n n
1        1        pron v art n n
1        1        qu adj n v n n
Types = 56    Tokens = 79
Times used = 56   Times used * Frequency = 79
```

Semantics: $\{ a \mid (\exists \langle a,b,c \rangle \in [\text{verb}])$
$(b \in [\text{np2}] \wedge c \in [\text{np1}]) \}$

(As mentioned in Chapter 5, the numbers following the 'np' symbols in the above denotation indicate the order of the symbols in the utterance. If no numbers occur, then the order in the denotation is the same as the order in the string under examination. Recall that the use of set-language in giving the semantics of a string is formally an abbreviation since the formal notion is that of a LISP-type expression of arguments and functions. See Chapter 5.)

Rule (3,4) handles a case of a verb phrase where the first noun-phrase following the verb is the indirect object, and the second is the direct object of the verb. Recall that verbs are a subset of

$$D^3 \cup D^2 \cup D$$

and that the verb therefore, if it takes both direct and indirect objects, will have as elements ordered triples of the form

<subject,direct-object,indirect-object> .

Many of these utterances are incorrectly described by this semantic rule. Very frequently, more subtle markings are needed in the dictionary to indicate how many objects the verbs may take. Many words (such as 'apple', 'alphabet') are classed only as nouns, while they are clearly used as adjectives in some utterances involved here.

The following utterances are all incorrectly described by the semantics, whereby

$$15\vartheta[\text{persp v n n}] =$$

$$\text{if } [\text{persp}] \subseteq \{ a \mid (\exists \langle a,b,c\rangle \in [v]) \; (b \in [n2] \wedge c \in [n1]) \}$$

then TRUE else FALSE

The utterances represented are:

(From: persp v n n)

2 it goes duck, duck.
he goes meow, meow.
he says moo, moo.
i buy apple juice.
it goes ding, ding.

(From: persp v,mod n n)

2 i want orange juice.
2 it go ding, ding.
i want alphabet cereal.

(From: persp v,aux n n)

i have bubble gum.
she has baby lizards.
we have syrup pot.
you have coffee cake.

However, other utterances are correct, as in

$$4\vartheta[\text{persp v persp n}] =$$

$$\text{if } [\text{persp}] \subseteq \{ a \mid (\exists \langle a,b,c\rangle \in [v]) \; (b \in [n] \wedge c \in [\text{persp}]) \} .$$

then TRUE else FALSE

which represents

(From: persp v persp n)

he brings me toys.
i gave him crackers.
i put it back.

(Remark: There is clearly a dictionary error on the word 'back'.)

(From: persp v,aux persp n)

i do it kitty.

(Remark: Here, the order of objects is inverted.)

Also, examine

2 persp aux v persp art n

(From: persp#aux,persp#link v persp art n,v)

2 he's giving him a kiss.

which are correctly analyzed.

Let me stress the following point. While I have found many cases that do not work in this semantics and consequently am forced to say that it needs reworking, the methods of lexical disambiguation used are often strikingly impressive. Notice the above utterances deriving from 'persp#aux,persp#link v persp art n,v'. This lexical form represents four alternatives. Only one of these is recognized by GE1; it is the correct representation, we would agree. I am personally convinced that more subtle dictionaries and grammars can solve the problems of disambiguation at a surface level in more cases than might have previously been thought possible.

Negating words ('neg') can occur in conjunction with rule (3,4). An example is

```
10[persp v neg n n] =
        if [persp] ⊆ { a | ( ∃<a,b,c>  ∈
                          (D^3 U D^2 U D) ~ [v])
                          (b ∈[n2]  ∧   c ∈ [n1] ) }
                then TRUE else FALSE .
```

The utterance involved is

he has no back seat

so the denotation is incorrect in this case: the word 'no' is here a quantifier, and 'back#seat' should be a noun.

<u>(3,5) vp -> verb prepp np</u>

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
2	1	persp v prep art n n	
2	1	persp v prep pronadj n n	
2	1	persp aux v prep art n n	
1	1	aux persp v prep n n	
1	1	conj persp v prep n n	
1	1	mod persp v prep pn pn	
1	1	mod persp v prep art n n	
1	1	mod persp v prep pronadj n n prep art n	
1	1	neg mod persp v prep persp n	

```
1        1        persp v prep pn pn
1        1        persp v prep pron art n
1        1        persp v prep art n adj n
1        1        persp v prep pronadj n pn
1        1        persp aux v prep persp padj n
1        1        persp mod neg v prep art pron n
1        1        persp v prep art n adj n prep pronadj adj n
1        1        pn v prep pn n
Types = 17    Tokens = 20
Times used = 17   Times used * Frequency = 20
```

Semantics: $\{ a \mid (\exists \langle a,b,c\rangle \in [\text{verb}])$
$(b \in [\text{np}] \wedge c \in [\text{prepp}]) \}$

The prepositional phrase ('prepp') is the indirect object of the verb, and the noun-phrase is the direct object. For example,

```
2@[persp v prep art n n] =
        if $[\text{persp}] \subseteq \{ a \mid ( \exists \langle a,b,c\rangle \in [v])$
            $(b \in [n] \wedge$
            $c \in \{ a \mid ( \exists \langle a,b\rangle \in [\text{prep}])$
                $(b \in \text{QUANTIF}([\text{art}],[n]))\} )\}$

        then TRUE else FALSE
```

represents the utterances

(From: persp v prep art n n)
he get over the tape recorder

(Remark: Dictionary error:
'tape#recorder' should be a noun)

(From: persp v,mod prep,adv art n n)
he go in the bath tub

(Remark: Dictionary error:
'bath#tub' should be a noun)

Also, consider the utterances representing

2 persp v prep pronadj n n

which are

i eat with my mommy hamburger.
you sit on my suit pants.

(Remark: 'suit#pants' should be a noun)

Most of the applications of rule (3,5) seem to be failures.

(3,6) vp -> verb np prepp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
14	2	persp v pron prep pron	
5	2	persp v persp prep art n	
4	2	mod persp v pron prep pron	
3	2	persp mod v pron prep pron	
3	2	persp mod v persp prep persp	
2	2	persp v n prep persp	
2	2	persp v persp prep pn	
2	2	persp v pron prep art n	
2	2	persp v art n prep persp	
2	2	persp v persp prep persp	
2	2	persp v n prep pronadj n	
2	2	persp v adj n prep persp	
2	2	persp v pron prep pronadj n	
2	2	persp mod v art n prep persp	
2	2	persp v persp prep pronadj n	
2	2	persp aux v n prep pronadj n	
2	2	persp mod neg v art n prep pronadj n	
1	2	conj persp v pron prep pron	
1	2	int persp v pron prep pron	
1	2	mod persp v pron prep n	
1	2	mod persp v pron prep art n	
1	2	mod persp v persp prep qu n	
1	2	mod persp v pronadj n prep n	
1	2	mod persp v persp prep art n	

```
1         2         mod persp v pronadj n prep pron
1         2         n pn aux v n prep art n
1         2         n v art n prep persp
1         2         n v pron prep persp
1         2         n v persp prep persp
1         2         n v pronadj n prep art n
1         2         n v pronadj n prep persp
1         2         neg persp mod neg v pron prep pron
1         2         persp v art n prep n
1         2         persp v n prep art n
1         2         persp mod v pron prep n
1         2         persp v art n prep pron
1         2         persp v qu n prep persp
1         2         persp mod neg v n prep n
1         2         persp aux v n prep persp
1         2         persp v pron prep padj n
1         2         persp v pron prep qu n aux
1         2         persp mod v persp prep pron
1         2         persp mod v persp prep art n
1         2         persp v art n prep art adj n
1         2         persp mod neg v n prep persp
1         2         persp v art adj pron prep pron
1         2         persp aux v qu pron prep persp
1         2         persp mod v pronadj n prep persp
1         2         persp mod v persp prep pronadj n
1         2         persp mod neg v adj n prep persp
1         2         pn n mod neg v pron prep pron
1         2         pn v pron prep persp
1         2         pronadj n v art n prep persp
Types = 53    Tokens = 89
Times used = 53   Times used * Frequency = 89
```

Semantics:

$$\{ a \mid (\exists \langle a,b,c \rangle \in [verb]) \; (b \in [np] \wedge c \in [prepp]) \}$$

Notice that the forms using rule (3,6) are all grammatically ambiguous. The ambiguity is whether or not the prepositional phrase is an object of the verb or a modifier of the noun phrase preceding it. See the discussion of grammatical ambiguity in Section 2 below.

(3,8) vp -> verb np prep

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
14	1	persp v persp prep	
12	1	persp v pronadj n prep	
10	1	persp mod v persp prep	
6	1	persp mod neg v persp prep	
3	1	n v persp prep	
3	1	persp v art n prep	
3	1	persp mod v pron prep	
3	1	persp mod neg v pronadj n prep	
2	1	mod persp v persp prep	
2	1	persp v n prep	
2	1	persp v pron prep	
2	1	persp mod neg v art n prep	
1	1	art n v persp prep	
1	1	conj persp v art n prep	
1	1	conj persp v pronadj n prep	
1	1	conj persp mod v art n prep	
1	1	conj persp mod neg v persp prep	
1	1	int pn v pronadj n prep	
1	1	int persp mod v persp prep	
1	1	n mod v persp prep	
1	1	n persp mod v persp prep	
1	1	n v n prep	
1	1	persp v qu n prep	
1	1	persp mod v art n prep	
1	1	persp mod neg v n prep	
1	1	persp mod v pronadj n prep	
1	1	pn v n prep	
1	1	pn v persp prep	
1	1	pron v neg qu n prep	

Types = 29 Tokens = 79
Times used = 29 Times used * Frequency = 79

Semantics: { a | (∃ <a,b> ∈

[COMBINE([verb] ,PREP)])

(b ∈ [np]) }

The preposition is taken to be a part of the meaning of the verb, and hence, the function COMBINE is used. Consider the utterances represented by

14 persp v persp prep

(From: 11 persp v persp prep,adv)

3 i dum it out.
i cover them up.
i covered them up.
i eat em up.
i eat him up.
i get it out.
i pushing it up.
i take it out.
you pull them up.

(From: 2 persp v persp prep)

he shave it off.
i turn it on.

(From: 1 persp v,aux persp prep,adv)

i do them up.

The function associated with (3,8) is apparently reasonable.

(3,9) vp -> verb prepp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
13	1	mod persp v prep persp	
12	1	persp v prep persp	
9	1	persp v prep pronadj n	
8	1	pers v prep art n	
7	1	persp v prep pron	

7	1	persp mod v prep persp
7	1	persp aux v prep art n
6	1	persp v prep qu n
5	1	mod persp v prep n
4	1	persp aux v prep persp
3	1	mod persp v prep pron
3	1	mod persp v prep pronadj n
3	1	persp aux v prep n
3	1	persp mod v prep pron
3	1	persp aux v prep pronadj n
2	1	persp v prep n
2	1	persp v prep pn
2	1	persp v prep padj n
2	1	persp aux v prep qu n
2	1	persp mod v prep pronadj n
2	1	persp mod neg v prep pronadj n
1	1	adj n v prep qu n
1	1	art n v prep art n
1	1	art n conj art n v prep art n
1	1	conj art n v prep pron
1	1	conj pn v prep qu pron
1	1	conj persp v prep art n
1	1	conj art n v prep persp
1	1	conj pn mod v prep persp
1	1	conj persp aux v prep pron
1	1	conj persp mod v prep art n
1	1	conj persp aux v prep adj n
1	1	conj pn conj pn aux v prep n
1	1	conj persp persp mod v prep pronadj n
1	1	int persp v prep pronadj n
1	1	mod n v prep art n
1	1	mod pron v prep art n
1	1	mod persp v prep qu n
1	1	mod persp v prep art n
1	1	n mod v prep art adj n
1	1	n n mod v prep pron
1	1	n n v prep pronadj n prep persp
1	1	n persp v prep n
1	1	n persp aux v prep art n
1	1	n pn mod neg v prep art n
1	1	n v prep qu n
1	1	n v prep art n
1	1	n v prep persp
1	1	n v prep pronadj n
1	1	neg n pn v prep pronadj n
1	1	neg persp aux v prep persp
1	1	neg persp mod v prep pronadj n
1	1	persp mod v prep n
1	1	persp v prep adj n
1	1	persp mod v prep pn

```
1        1        persp aux v prep pron
1        1        persp mod v prep qu n
1        1        persp mod neg v prep n
1        1        persp mod v prep padj n
1        1        persp v prep n prep pron
1        1        persp v prep art adj pron
1        1        persp mod neg v prep pron
1        1        persp v prep persp prep n
1        1        persp mod v prep adj pron
1        1        persp aux v prep n prep n
1        1        persp mod neg v prep persp
1        1        persp mod neg v prep art n
1        1        persp aux neg v prep art n
1        1        persp v prep art n conj art n
1        1        persp mod v prep pron prep pron
1        1        pn mod v prep persp
1        1        pron v prep pn
1        1        pron padj n v prep n
1        1        pron v prep pronadj n
1        1        pron mod v prep persp
1        1        qu n v prep pronadj n
1        1        qu n mod v prep pronadj n
1        1        qu pron v prep n
Types = 78    Tokens = 162
Times used = 78   Times used * Frequency = 162
```

Semantics: $\{ a \mid (\exists \langle a,b,c\rangle \in [verb])$

$(c \in [prepp]) \}$

This rule is intended to be used when 'prepp' is an indirect object to the verb, and the verb is missing.

Example:

```
13      mod persp v prep persp

(From:   9      mod persp v prep persp  )

3       let me talk to it.
2       let me listen to it.
        can i talk to it?
        can i talk to him?
        can i listen to it?
        may i talk to it?
```

(Remark: The above seem to reinforce my interpretation.)

(From: 2 mod persp v,mod prep persp)

can i go with him?
let me go with you.

(Remark: Here, the prepositional phrase is adverbial, so my semantics is incorrect.)

(From: 1 mod persp v,mod prep,adv persp)

can i go in it?

(Remark: Again, an adverbial phrase.)

(From: 1 mod#persp v prep,adv persp)

lemme talk in it.

The function for (3,9) is only partly successful. Notice, however, that GE1 does correctly disambiguate in the above utterances.

(11,1) verb -> v

Types = 576 Tokens = 2497
Times used = 642 Times used * Frequency = 2604

Semantics: [v]

(11,2) verb -> v neg

TERMINAL FORMS

Types No. of Derivations Form Times rule used on form (If different from 1)

```
------------------------------------------------------------
6        1        persp v neg
6        1        persp v neg mod
2        1        v neg
1        1        art n mod neg v neg
1        1        conj n pn v neg
1        1        conj pron v neg
1        1        conj persp mod neg v neg
1        2        conj mod qu n prep n v neg persp        2
1        1        n aux v neg persp
1        1        persp v neg n
1        1        persp v neg n n
1        1        persp v neg adj n
1        1        pn v neg
1        1        pron v neg
1        1        pron v neg qu n prep
1        1        qu n v neg persp
Types = 16    Tokens = 27
Times used = 17   Times used * Frequency = 28
```

Semantics: (D^3 U D^2 U D) ~ [v]

Rule (11,2) does not work correctly when used with rule (3,2). The only form using both rules (3,2) and (11,2) is

1   pron v neg qu n prep

representing

1   this has not two children in.

Apart from the fact of the strangeness of this utterance, notice that the semantics gives this denotation:

if [pron] ⊆

{a|(∃ <a,b> ∈ [COMBINE(((D^3 U D^2 U D) ~ [v]),prep]))

(b ∈ QUANTIF([qu],[n])))}

then TRUE else FALSE .

This denotation fails to COMBINE the preposition with the verb until after the denotation of the verb has been computed. A more reasonable denotation is

```
     if [pron] ⊆
   {a|(∃ <a,b> ∈ ((D^3 U D^2 U D)  ~ [COMBINE(v,prep)]))
               (b ∈ QUANTIF([qu],[n]))}
               then TRUE else FALSE .
```

This is, however, a relatively minor problem to fix.

(19.1) linkp -> link

```
Types = 178    Tokens = 860
Times used = 182            Times used * Frequency = 865

Semantics:       [link]
```

(19.2) linkp -> link neg

```
                       TERMINAL FORMS

Types   No. of           Form     Times rule used on form
     Derivations                  (If different from 1)
-------------------------------------------------------------

4          1        persp link neg adj
3          1        persp link neg art n
3          1        persp link neg adv adj
2          1        persp link neg n
2          1        persp link neg qu adj
```

```
2        1        persp link neg art adj n
2        1        persp link neg prep persp
2        1        pron link neg n
2        1        pron link neg art n
1        1        conj pron link neg adj
1        1        conj persp link neg adj
1        1        link neg persp adj
1        1        neg persp link neg adj
1        1        neg pron link neg art pn
1        1        persp link neg n n
1        1        persp link neg adj adj adj
1        1        persp link neg art adv adj n
1        1        persp link neg qu adj adj adj
1        1        pn link neg adj
1        1        pron link neg adj
1        1        pron link neg adv adj
1        1        pron link neg qu pron
1        1        pron link neg art adj n
Types = 23    Tokens = 36
Times used = 23   Times used * Frequency = 3o

Semantics:        [link]
```

9. RULES FOR NOUN-PHRASES THAT STAND ALONE

The nom and nom1 rules add nothing to the semantical understanding of ERICA. Rather, they account for the observation that the generation of noun-phrases that stand alone seems to be different from the generation of noun-phrases that stand with predicates.

<u>(7.1) nom -> npsub prepp</u>

TERMINAL FORMS

```
Types   No. of            Form    Times rule used on form
     Derivations                  (If different from 1)
-------------------------------------------------------------
```

```
7         1        n prep art n
6         1        pron prep persp
5         1        pron prep pron
4         1        pron prep n
3         1        adj n prep persp
3         1        n prep pn
3         1        n prep persp
2         1        art n prep persp
2         1        n prep n
2         1        pn prep art n
2         1        pron prep pn
1         1        adj adj n prep art n
1         1        adv adj n prep pronadj n
1         1        conj pron prep pron
1         1        persp prep persp
1         1        pn prep pn conj persp
1         1        qu pron prep pronadj n
Types = 17    Tokens = 45
Times used = 17   Times used * Frequency = 45

Semantics:         [npsub] ∩ [prepp]
```

(7.3) nom -> npsub conj npsub

TERMINAL FORMS

```
Types   No. of            Form    Times rule used on form
     Derivations                  (If different from 1)
-------------------------------------------------------------

6         1        art n conj art n
6         1        n conj n
4         1        n conj art n
3         1        pron conj pron
2         1        n conj persp
2         1        pn conj pn
2         1        pn conj art n
1         1        adj n conj n
1         1        conj n conj n
1         1        n conj pn
1         1        n conj pron
1         1        neg art n conj art n
1         1        neg pron conj pron
1         1        persp conj pn
1         1        persp conj persp
1         1        pn conj n
```

```
1        1        pn conj persp
1        1        pn conj pronadj n
1        1        pronadj adj n conj art n
1        1        pronadj n conj pronadj n
Types = 20    Tokens = 38
Times used = 20   Times used * Frequency = 38


Semantics:       ( [npsub] ) U  ( [npsub] )
```

(7.4) nom -> nom1

```
Types = 117    Tokens = 1343
Times used = 118          Times used * Frequency = 1344


Semantics:       [nom1]
```

(7.5) nom -> qadp

TERMINAL FORMS

```
Types   No. of           Form    Times rule used on form
     Derivations                 (If different from 1)
-----------------------------------------------------------


66       1        adj
10       1        pronadj
7        1        adj adj adj
6        1        adj adj
6        1        neg adj
6        1        qu adj
5        1        adv adj
4        1        padj
3        1        art adj
2        1        art
2        1        neg adj adj
1        1        adj adj adj adj adj
1        1        adj adj adj adj adj adj adj adj
1        5        adv adv adj adj 5
1        1        art adj adj
1        1        conj pronadj adv adj
1        1        int adv adj
```

1	1	neg adv adj
1	1	pronadj adj

Types = 19   Tokens = 125
Times used = 23   Times used * Frequency = 129

Semantics: [qadp]

(18,1) nom1 -> npsub

Types = 117   Tokens = 1343
Times used = 118   Times used * Frequency = 1344

Semantics: [npsub]

(18,2) nom1 -> nom1 npsub

Types = 67   Tokens = 264
Times used = 170   Times used * Frequency = 431

Semantics: [nom1] ∩ [npsub]

10. RULES GENERATING SENTENCES

The a-rules generate complete sentences. The adding of interjections, conjunctions, plus the joining two sentences together into one utterance, are accomplished by the s-rules.

(4,1) a -> nom

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
553	1	n	
92	1	art n	
90	1	n n	
89	1	pron	
66	1	adj	
55	1	adj n	
43	1	pn	
34	1	pronadj n	
30	1	qu pron	
29	1	qu n	
18	1	neg n	
17	1	persp	
17	1	pn n	
15	1	int n	
11	1	adj adj n	
11	1	n n n	
11	1	pron art n	
10	1	int n n	
10	1	pronadj	
8	1	art adj n	
8	1	art n n	
8	1	conj art n	
8	1	pn pn	
7	1	adj adj adj	
7	1	conj n	
7	1	n n n n n	
7	1	n pn	
7	1	n prep art n	
6	1	adj adj	
6	1	adj pron	
6	1	art n conj art n	
6	1	conj pn	
6	1	n conj n	
6	1	neg adj	
6	1	pron qu pron	
6	1	pron prep persp	
6	1	qu adj	
5	1	adv adj	
5	1	neg art n	
5	1	pron prep pron	
4	1	adj n n	
4	1	conj n n	
4	1	n conj art n	
4	1	neg n n	
4	1	padj	
4	1	pron prep n	

4	1	qu n n
3	1	adj n prep persp
3	1	adj pn
3	1	adv adj n
3	1	art adj
3	1	conj pron
3	1	conj persp
3	1	conj art adj n
3	1	int pn
3	1	n int
3	1	n n n n
3	1	n persp
3	1	n prep pn
3	1	n prep persp
3	1	neg pron
3	1	persp n
3	1	pron conj pron
3	1	qu adj n
3	1	qu adj n n
3	1	qu pn
3	1	qu pron qu pron
3	1	qu pron qu pron qu pron
2	1	art
2	1	art n prep persp
2	1	n conj persp
2	1	n n n n n n n n n
2	1	n n n n n n n n n n n n
2	1	n prep n
2	1	neg adj adj
2	1	neg qu n
2	1	persp n persp
2	1	pn conj pn
2	1	pn conj art n
2	1	pn pn n
2	1	pn prep art n
2	1	pron qu n
2	1	pron prep pn
2	1	pronadj n pronadj n
2	1	qu n pron
2	1	qu pron qu pron pron
1	1	adj adj n n
1	1	adj adj pron
1	1	adj adj adj n
1	1	adj adj adj adj adj
1	1	adj adj n prep art n
1	1	adj adj adj adj adj adj adj adj
1	1	adj n pn
1	1	adj n int
1	1	adj n adj n
1	1	adj n conj n

1	1	adj pron adj pron
1	2	adv adv adj n 2
1	5	adv adv adj adj 5
1	1	adv adj n prep pronadj n
1	1	aff n
1	1	art ad adj
1	1	art adj n n
1	1	art adj pron
1	1	art adj adj n
1	1	art adj adj pron
1	1	art adj adj adj n
1	1	art n n n n
1	1	conj qu n
1	1	conj art n n
1	1	conj n conj n
1	1	conj pronadj n
1	1	conj art adj adj n
1	1	conj pron prep pron
1	1	conj pronadj adv adj
1	1	int adv adj
1	1	int n pn
1	1	int n adj n
1	1	int n n n n
1	1	int pron
1	1	int persp
1	1	int pronadj n
1	1	int pron qu pron
1	1	n adj n
1	1	n conj pn
1	1	n conj pron
1	1	n n n n n n
1	1	n n n n n n n
1	1	n n n n n n n n n n n n n n
1	1	n n n n n n n n n n n n n n n n n
1	1	n n n n persp n n n n n n n n n n n n n n
1	1	n n pn
1	1	n padj n
1	1	n pron
1	1	n pronadj n n
1	1	n qu n
1	1	n qu pron
1	1	neg adj n
1	1	neg adv adj
1	1	neg art n conj art n
1	1	neg n pn
1	1	neg pn
1	1	neg pron conj pron
1	1	padj n
1	1	persp n n
1	1	persp conj pn

1	1	persp adj pron
1	1	persp conj persp
1	1	persp prep persp
1	1	persp art adj adj n
1	1	pn art n
1	1	pn conj n
1	1	pn conj persp
1	1	pn conj pronadj n
1	1	pn n pn n pn n
1	1	pn pn pn pn
1	1	pn prep pn conj persp
1	1	pron persp
1	1	pron art pron
1	1	pron art adj n
1	1	pron art adj adj n
1	1	pronadj adj
1	1	pronadj pron
1	1	pronadj n n n
1	1	pronadj adj n
1	1	pronadj adj n conj art n
1	1	pronadj n conj pronadj n
1	1	qu adj adj n
1	1	qu n qu n qu n qu n qu n qu n
1	1	qu pron prep pronadj n
1	1	qu pron qu pron pron qu pron
1	1	qu pron qu pron qu pron conj
1	1	qu pron qu pron qu pron qu n

Types = 173   Tokens = 1551
Times used = 178   Times used * Frequency = 1556

Semantics: [nom]

Out of 9,085 utterances in ERICA, recall that 7,048 were recognized by GE1. Of these, 1,551 are noun-phrases that stand alone, as generated by the rule (4,1). Because of the interest in this class, I have included above all the forms.

(4,2) a -> inter

Types = 1   Tokens = 7
Times used = 1   Times used * Frequency = 7

Semantics: IMMED ∩ [inter]

The utterances using (4,2) are:

4 what?
2 what.

(Remark: Presumably the utterance 'what.' should be a question.)

1 who?

The 'inter' words are the interrogative pronouns. The denotation of an 'inter' is the set of things in D that could satisfy the word. For example, [what] is the set of inanimate objects, and [who] is the set of animate (perhaps sentient) objects. The semantics for the rule says to intersect [inter] with IMMED. I think this is a reasonable approximation.

(4,3) a -> subj vbl

Types = 380 Tokens = 1598
Times used = 424 Times used * Frequency = 1675

Semantics:

```
    IF ( [subj] )  ⊆  ( [vbl] )
        THEN TRUE ELSE FALSE
```

(4,4) a -> inter vbl

```
                         TERMINAL FORMS
Types   No. of            Form    Times rule used on form
     Derivations                  (If different from 1)
-------------------------------------------------------------

5        1        inter aux v prep
3        1        inter v
1        1        inter v persp
Types = 3    Tokens = 9
Times used = 3    Times used * Frequency = 9
```

Semantics: [inter] ∩ [vbl] ∩ IMMED

For example,

5@[inter aux v prep] =

[inter] ∩ [aux] ∩

[COMBINE([v],PREP)] ∩ [IMMED]

represents

```
(From:   3      inter#aux,inter#link v,mod prep)
3        what's going on?
(From:   2      inter#aux,inter#link v prep,adv)
2        what's happening outside?
```

(Remark: Here, lexical disambiguation by GE1 has chosen that 'outside' is a preposition; it is more correctly an adverb.)

Rule (4,4) seems reasonably successful.

(4,5) a -> subj linkp prepp

```
                              TERMINAL FORMS

Types   No. of            Form    Times rule used on form
     Derivations                  (If different from 1)
-------------------------------------------------------------

4        1         persp link prep art n
4        1         pron link prep pn
4        1         pron link prep persp
2        1         persp link neg prep persp
2        1         persp link prep pronadj n
1        1         art n link prep persp
1        1         conj persp link prep art n
1        1         int persp link prep persp
1        1         n link prep persp
1        1         persp link prep n
1        1         persp link prep pron
1        1         persp link prep art pron
1        1         persp link prep art adj n
1        1         pn link prep art n
1        1         pron link prep art n
1        1         pron link prep pronadj n
Types = 16    Tokens = 27
Times used = 16   Times used * Frequency = 27
```

Semantics:

IF ([subj]) ⊆ (AUXFCN([linkp] , [prepp]))
THEN TRUE ELSE FALSE

An interesting case involving the negating particle 'neg' is:

20[persp link neg prep persp] =

if [persp] ⊆

(AUXFCN([link neg] ,

{ a | (∃<a,b> ∈ [prep])

(b ∈[persp]) })

then TRUE else FALSE

representing

```
(From:   2      persp#aux,persp#link neg prep persp)

2       it's not for me.
```

This is not implausible.

(4.6) a -> inter linkp

```
Types = 1    Tokens = 3
Times used = 1    Times used * Frequency = 3

Semantics:       [inter] ∩ AUXFCN( [linkp]  , IMMED )
```

(4.7) a -> mod subj

```
Types   No. of          Form    Times rule used on form
     Derivations                (If different from 1)
----------------------------------------------------------

5        2       mod persp
2        2       mod pron
1        2       mod pronadj n
1        2       neg mod persp
Types = 4    Tokens = 9
Times used = 4    Times used * Frequency = 9

Semantics:
        IF ( [subj] )   ⊆  ( [mod] )
            THEN TRUE ELSE FALSE
```

(4.8) a -> prepp

TERMINAL FORMS

```
Types   No. of          Form    Times rule used on form
     Derivations                (If different from 1)
----------------------------------------------------------
```

```
30       1        prep art n
18       1        prep n
13       1        prep pronadj n
9        1        prep persp
5        1        prep pn
3        1        prep pron
3        1        prep padj n
1        1        aff prep n prep persp
1        1        int prep persp
1        1        int prep art n
1        1        neg prep qu n
1        1        neg prep persp
1        1        neg prep padj n
1        1        neg prep pronadj n
1        1        prep adj n
1        1        prep art adj n
1        1        prep pn conj pn
1        1        prep pronadj n conj n
Types = 18    Tokens = 92
Times used = 18   Times used * Frequency = 92
```

```
Semantics:       [prepp]
```

<u>(4.9) a -> linkp subj qadp</u>

TERMINAL FORMS

```
Types   No. of            Form    Times rule used on form
     Derivations                  (If different from 1)
-------------------------------------------------------------

4        1        link persp adj
1        1        link pron qu
1        1        link pron adj
1        1        link neg persp adj
Types = 4    Tokens =
Times used = 4    Times used * Frequency = 7
```

```
Semantics:
        If ( [subj] )  ⊆  ( AUXFCN( [linkp] ,[qadp] ) )
            then TRUE else FALSE
```

Consider, for example,

```
40[link persp adj] =
        if [persp] ⊆
           (AUX-CN( [link] , [qadp] ))
            then TRUE else FALSE
```

representing

```
(From:   4      link,aux persp adj)

2       are they blue?
        are they good?
        is it warm?
```

Notice that all these utterances are questions. Since, by convention, the meaning of a question is its answer, the semantics works correctly.

One can explain the apparently puzzling claim that the meaning of a question is its answer by allowing that Erica will understand the structure of her data base (the model 𝔄) without necessarily knowing all the details of that data base.

Of course, questions are different from declarative statements in that they require a different response from the other party(ies), but this is no problem.

<u>(4,10) a -> linkp subj np</u>

TERMINAL FORMS

```
Types   No. of          Form      Times rule used on form
     Derivations                  (If different from 1)
---------------------------------------------------------

5        1        link pron art n
2        1        link persp n
2        1        link pron persp
2        1        link persp art n
2        1        link pron pronadj n
Types = 5    Tokens = 13
Times used = 5    Times used * Frequency = 13
```

Semantics:

```
          IF ( [subj] )  ⊆  ( AUXFCN( [linkp] , [np] ) )
               THEN TRUE ELSE FALSE
```

The intended interpretation is that 'subj' is the subject, and that 'np' is a predicate nominative. Notice that no utterance uses 'link neq', which is a possibility in grammar GE1.

5@[link pron art n] =

```
          if [pron]  ⊆  AUXFCN([link],QUANTIF([art],[n]))
          then TRUE else FALSE
```

represents

(From: 5   link,aux qu,pron art n)

```
1         is this a mom?
1         is that a rat?
1         is that a man?
1         is this a daddy?
1         is that a pumpkin?
```

This is a plausible interpretation for these utterances, which are all questions.

(4,11) a -> subj linkp np

Types = 76 Tokens = 342
Times used = 76 Times used * Frequency = 342

Semantics:

```
        IF ( [subj] ) ⊆
              ( AUXFCN( [linkp] , [np] ) )
        THEN TRUE ELSE FALSE
```

Here, 'subj' is again the intended subject, and 'np' the predicate nominative.

Consider

30[persp link neg art n] =

```
        if [persp]  ⊆
            AUXFCN([link neg], QUANTIF( art],[n]))
        then TRUE else FALSE
```

which represents

(From: 2 persp link,aux neg art n)

1 he is not a puppet.
1 i am not a bear.

(From: 1 persp#aux,persp#link neg art n)

1 i'm not a girl.

(4,12) a -> subj linkp gadp

Types = 39 Tokens = 132
Times used = 41 Times used * Frequency = 135

```
Semantics:
        IF ( [subj] )  ⊆  ( AUXFCN( [linkp]  , [qadp] ) )
            THEN TRUE ELSE FALSE
```

The 'qadp' is a predicate adjective phrase in rule (4,12).

(4.13) a -> auxilp subj vp

```
Types = 64    Tokens = 181
Times used = 72   Times used * Frequency = 192
```

```
Semantics:
        IF ( [subj] )  ⊆  ( AUXFCN( [auxilp]  ,  [vp] ) )
            THEN TRUE ELSE FALSE
```

(4.14) a -> subj np vbl

```
Types = 43    Tokens = 55
Times used = 45   Times used * Frequency = 57
```

```
Semantics:
        IF ( [subj] )  ⊆
               { a | ( ∃<a,b> ∈  [vbl] ) ( b  ∈ [np] ) }
        THEN TRUE ELSE FALSE
```

(4.15) a -> subj linkp np np

TERMINAL FORMS

```
Types   No. of          Form    Times rule used on form
     Derivations                (If different from 1)
-------------------------------------------------------

19       1       pron link art n n
6        1       pron link n n
```

```
3       1       persp link n n
3       1       pron link art pn n
2       1       pron link art n pron
2       1       pron link pronadj n n
2       1       pronadj n link n n
1       1       conj pron link art n n
1       1       conj pron link art pn n
1       1       conj persp link art adj n n
1       1       neg pron link pn n
1       1       neg persp link art pn n
1       1       persp link n qu n
1       1       persp link neg n n
1       1       persp link art n n
1       2       persp link adv adv adj pron n    2
1       1       pron link pn n
1       1       pron link pron art n
1       1       pron link persp n conj
1       1       pron link pn pn conj pn
1       1       pron link pron qu adj n
1       1       pron link art adj pron art n
Types = 22    Tokens = 52
Times used = 23   Times used * Frequency = 53
```

```
Semantics:
        IF ( [subj] ) ⊆

                (AUXFCN( [linkp] , ( [np] ∩ [np] ) ))

              THEN TRUE ELSE FALSE
```

The intended semantics is based on the assumption that the two noun-phrases are in apposition. Consider the utterances represented by

```
19      pron link art n n
```

some of which are

```
(From:   18      pron#aux,pron#link art n n)

4       there's a kitty cat.
2       there's a tape recorder.
        that's a tea pot.
        that's a music cat.
```

Notice that the apposition interpretation is not contradicted, although some combinations should be listed as single words (such as 'kitty#cat', 'tape#recorder'.) Moreover, 'there's' and 'that's' and similar demonstrative phrases should be given a better classification than 'pron#aux,pron#link'.

(4.16) a -> auxilp subj np

```
                         TERMINALS
-------------------------
1        1        aff mod persp n
1        1        mod art n n
1        1        mod neg n pron
1        1        mod neg persp n
1        1        mod persp n

Types = 5    Tokens = 5
Times used = 5    Times used * Frequency = 5
```

Semantics:

If $([subj]) \subseteq \{ a \mid (\exists \langle a,b \rangle \in$

$\mathrm{AUXFCN}([auxilp] , \mathrm{IMMED})) (b \in [np]) \}$

then TRUE else FALSE

The intention is that these utterances are missing their main verbs. Consider

1 mod art n n

which represents

maybe the milk man.

Here it is plausible that the main verb is missing but assumed as a part of the 'context'. It is quite possible that this semantics should have several contextual parameters, representing, say, objects, properties, actions, under immediate consideration. I have used only the set IMMED to indicate the presence of a contextual parameter. The idea of extending this to several contextual parameters is straightforward. The implementation may be rather involved and is beyond the scope of this work.

(4.19) a -> auxilp subj

```
Types = 12     Tokens = 38
Times used = 14   Times used * Frequency = 40

Semantics:
        IF ( [subj] )  ⊆  ( AUXFCN( [auxilp]  ,  IMMED ) )
            THEN TRUE ELSE FALSE
```

(4.20) a -> verb

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
229	1	v	
3	1	int v	
3	1	neg v	
2	1	v int	
2	1	v neg	
1	1	v aff	

Types = 6   Tokens = 240
Times used = 6   Times used * Frequency = 240

Semantics: if IMMED $\subseteq$ [verb]
then TRUE else FALSE

In these utterances the verb stands alone. For

229   v

the utterances are a simple verb. Examples:

70   lookit.

(Remark: Probably an imperative.)

70   know.

(Remark: Short for 'i don't know', according to the contexts.)

21   see.

The function for (4,20) works in many cases; 'lookit' and 'know' are notable failures.

Moreover, two utterances contain a negating particle:

3   neg v
2   v neg

For these utterances, it seems reasonable that the negating particle affects the verb. This semantics views these as being paired-denotation utterances, viz.:

[neg v] = <FALSE, [v] >

and hence the denotations given to these utterances are incorrect.

(4,21) a -> intadv auxilp subj vp

```
                    TERMINAL FORMS

Types   No. of           Form    Times rule used on form
     Derivations                 (If different from 1)
--------------------------------------------------------

2        1       intadv mod persp v
1        1       intadv aux qu n v
1        1       intadv aux persp v
1        1       intadv aux art n v
1        1       intadv aux pronadj n v
1        1       intadv mod neg persp v n
Types = 6    Tokens = 7
Times used = 6    Times used * Frequency = 7
```

Semantics: MEASURE(<auxilp#VP,INTADV>, ([SUBJ] ∩

AUXFCN([auxilp] , [vp])), [intadv])

The functions given for the interrogative adverbs are not well thought out. The utterances are questions, inquiring into such matters as 'where', 'when', or 'how' an action took place.

Consider

20[intadv mod persp v] =

MEASURE(<auxilp#VP,INTADV>, ([persp] ∩

AUXFCN([mod] , [v])) , [intadv])

representing

(From: 2 intadv#mod persp v,mod)

2 where'd it go?

(Remark: 'where'd' is here an 'intadv#aux'.)

The rule says:

1) Compute AUXFCN([did], [go]) . This gives us the set of all things that "did go".

2) Intersect this with [it].

3) Now, compute the adverbial function MEASURE on the arguments.

I leave the structure of adverbs in general and interrogative adverbs in particular as an unsolved problem.

(4.22) a -> intadv auxilp subj

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
5	1	intadv aux art n	
4	1	intadv aux pronadj n	
2	1	intadv aux n	
2	1	intadv aux pron	
2	1	intadv aux pronadj adj n	
1	1	intadv aux qu n	
1	1	intadv aux persp	
1	1	intadv aux art pron	
1	1	intadv aux qu adj n	
1	1	intadv aux art adj n	
1	2	intadv aux art n prep art n	2

Types = 11 Tokens = 21
Times used = 12 Times used * frequency = 22

Semantics: MEASURE(<auxilp#IMMED,INTADV>, [subj] ∩ AUXFCN([auxilp] , IMMED), [intadv])

A few examples:

```
(From:   4      intadv#aux,intadv#link art n)

1       where's a arrow?
1       where's an arrow?
1       where's the lady?
1       where's the buttons?

(From:   4      intadv#aux,intadv#link pronadj n)

1       where's my toys?
1       where's my door?
1       where's his sack?
1       where's my pillow?
```

(4,23) a -> intadv

```
                     TERMINAL FORMS

Types   No. of             Form    Times rule used on form
     Derivations                   (If different from 1)
-----------------------------------------------------------

3       1        intadv
Types = 1    Tokens = 3
Times used = 1    Times used * Frequency = 3


Semantics:        MEASURE(<IMMED,INTADV>,IMMED, [intadv])

        The utterances using (4,23) are:

1       how...
1       where?
1       why?
```

(4,24) a -> verb subj

```
Types = 31    Tokens = 251
Times used = 40   Times used * Frequency = 265
```

Semantics:

```
        IF ( [subj] )  ⊆  ( [verb] )
            THEN TRUE ELSE FALSE
```

(4,25) a -> advp subj auxilp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
27	1	adv persp aux	
24	1	adv persp mod	
4	1	int adv persp aux	
1	1	adv art n aux	
1	1	adv n aux	
1	1	conj adv persp aux	

Types = 6   Tokens = 58

Times used = 6   Times used * frequency = 58

Semantics:

```
        IF ( [subj] )  ⊆

        MEASURE( <auxilp,ADVP> ,

           AUXFCN( [auxilp]  ,  IMMED ) , [advp] )

            THEN TRUE ELSE FALSE
```

Some utterances using (4,25) follow.

27 adv persp aux

(From: adv persp link,aux)

10	there it is.
7	there he is.
5	there they are.
1	here he is.
1	here it is.
1	here we are.
1	here they are.
1	there we are.

These utterances represent a failure of lexical disambiguation. Here, the adverbs (all locatives) modify the linking verbs, but the grammar disambiguates to the auxiliary.

24 adv persp mod

(From: 24 adv persp v,mod)

7	here we go.
5	there you go.
4	here i go.
4	there we go.
1	here you go.
1	there i go.
1	there it go.
1	there they go.

Here the verb is an action verb, but the adverb doesn't modify at all. The words 'here' and 'there' act as interjections in the utterances.

4 int adv persp aux

(From: 4 int adv persp link,aux)

4 oh, there it is.

Again, the verb is not an auxiliary, so lexical disambiguation has failed.

(4.28) a -> subj auxilp

Types = 17 Tokens = 82
Times used = 17 Times used * Frequency = 82

Semantics:

```
        IF ( [subj] )  ⊆  AUxFCN( [auxilp] , IMMED )
              THEN TRUE ELSE FALSE
```

(4.29) a -> advp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
67	1	adv	
13	1	adv adv	
3	1	neg adv	
2	1	int adv	
1	1	adv adv adv	
1	1	conj adv	

Types = 6   Tokens = 87
Times used = 6   Times used * Frequency = 87

Semantics: [advp]

Some examples

(From: 55 adv)

29	here.
18	there.
2	tomorrow.
2	...here.
1	carefully.
1	down.
1	just.
1	there...

(From: 9 prep,adv adv)

3	in here.
2	in here.
2	under there.
1	in there.
1	out yonder.

(Remark: These adverbial phrases should probably be [illegible] as such in the dictionary, since they seem sufficiently unanalyzable. Alternatively, 'here', 'there', and 'yonder' could be thought of as nouns denoting places, as objects of the prepositions involved.)

(4.30) a -> inter subj

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
32	1	inter pron	
1	1	inter n	
1	1	inter qu n	
1	1	inter persp	
1	1	inter pronadj n	
1	2	inter pron prep art n	2

Types = 6 Tokens = 37
Times used = 7 Times used * Frequency = 38

Semantics: [inter] ∩ [subj] ∩ IMMED

Some examples:

(From: 32 inter qu,pron)

17	what that?
7	what this?
3	who that?
3	who this?
2	what those?

(4.31) a -> inter linkp subj

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
191	1	inter link pron	
18	1	inter link persp	
6	1	inter link qu n	
4	1	conj inter link pron	
3	1	inter link pronadj n	
2	1	inter link qu pron	
1	1	int inter link pron	
1	1	inter link art n	
1	1	inter link qu adj n	
1	2	inter link pron prep pronadj n	2

Types = 10 Tokens = 228
Times used = 11 Times used * frequency = 229

Semantics: [inter] ∩ [subj] ∩
AUXFCN([linkp] , [IMMED])

Examples:

(From: 157 inter#aux,inter#link qu,pron)

103	what's that?
36	what's this?
8	what's those?
5	who's that?
3	who's this?
1	what's ... this?
1	who's those?

(4,32) a -> inter np vbl

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
5	1	inter persp v	
2	1	inter persp v prep	
1	1	inter pron v	
1	1	inter persp mod v	
1	1	inter persp v qu n	

Types = 5 Tokens = 10
Times used = 5 Times used * Frequency = 10

Semantics:

$$[\text{inter}] \cap \{\, a \mid (\exists\langle a,b\rangle \in [\text{vbl}])(b \in [\text{np}]) \,\} \cap \text{IMMED}$$

Some utterances using (4,32):

(From: 2 inter persp v,aux)

1	what i have.
1	what she have.

(Remark: These do appear to be fragmentary, but instead of being main clauses simply missing a main verb, they seem to be subordinate clauses.)

(4,33) a -> advp subj vbl

Types = 7 Tokens = 26
Times used = 7 Times used * Frequency = 26

Semantics:

```
IF ( [subj] ) $\subseteq$
        MEASURE( <VBL,ADVP> , [vbl] , [advp] )
    THEN TRUE ELSE FALSE
```

Example:

(From: 15 adv persp v)

5	there he goes.
2	here i come.
2	here he goes.
2	there it goes.
1	here she goes.
1	there it fits.
1	there he stands.
1	wherever she goes.

(4.35) a -> vol subj prep

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
34	1	v persp prep	
5	1	v pron prep	
4	1	v art n prep	
2	1	v pronadj n prep	
1	1	int v persp prep	
1	1	mod neg v pronadj n prep	
1	1	neg v persp prep	
1	1	v n prep	
1	1	v persp n prep	
1	1	v pn prep	
1	1	v prep pronadj n prep	
1	1	v qu n prep	

Types = 12   Tokens = 53

Times used = 12   Times used * Frequency = 53

Semantics:

```
        IF ( [subj] ) ⊆ [COMBINE( [vol] ,PREP)]
            THEN TRUE ELSE FALSE
```

Examples:

(From: 20 v persp prep,adv)

7	turn it up.
3	eat me up.
2	pick it up.
2	pick them up.
1	eat it up.
1	eat them up.
1	put it away.
1	take it up.
1	take it out.
1	take him out.

(4.37) a -> verb subj np

Types = 21 Tokens = 38
Times used = 24 Times used * Frequency = 41

Semantics:

IF ([subj]) $\subseteq$

$$\{ a \mid (\exists \langle a,b \rangle \in [verb]) (b \in [np]) \}$$

THEN TRUE ELSE FALSE

The intended interpretation is that the 'subj' is a subject, and the 'np' is the direct object. Some mixed results follow.

(From: 9 v persp n)

2	did you, mommy.
2	thank you, mommy?
1	bring me curl.
1	drink it, doggie.
1	look it now.
1	make me fishy.
1	make me bubbles.

(From: 4 v art n n)

1	draw...a kitty cat.
1	see a tape recorder.
1	see the bunny rabbits.
1	tell the tape recorder.

Several of these are imperatives, with the 'subj' an indirect object; several others show nouns of direct address. The results of using this rule appear to be mixed.

(4,38) a -> intadv subj vbl

TERMINAL FORMS

```
Types   No. of          Form     Times rule used on form
     Derivations                 (If different from 1)
-------------------------------------------------------------

13       1        intadv persp v
1        1        intadv art n v
1        1        intadv persp v persp
1        1        intadv persp v art adj n
Types = 4    Tokens = 16
Times used = 4    Times used * Frequency = 16
```

Semantics: MEASURE(<VBL,INTADV>,[subj] ∩ [vbl], [intadv])

Some examples:

```
(From:   13      intadv persp v,mod)

6        where it go?
4        where they go?
1        where i go?
1        where you go?
1        where he going?
```

<u>(4.39) a -> auxilp v</u>

TERMINAL FORMS

```
------------------------
23       1       mod neg v
8        1       mod v
1        1       mod neg v int

Types = 3    Tokens = 32
Times used = 3    Times used * Frequency = 32
```

Semantics:

```
        IF ( IMMED ) ⊆ AUXFCN( [auxilp] , [v] )
            THEN TRUE ELSE FALSE
```

The intended interpretation is that the utterance is missing its subject. Some examples:

(From: 22 v#neg,mod#neg v)

22 don't know.

(From: 3 mod v)

2 wanna see.
1 wanna see?

(4.40) a -> advp linkp subj

Types = 12 Tokens = 34
Times used = 12 Times used * Frequency = 34

Semantics:

```
    IF ( [subj] ) ⊆

        MEASURE(<IMMED,ADVP>,

                AUXFCN([linkp],IMMED), [advp] )

        THEN TRUE ELSE FALSE
```

(4.41) a -> linkp qadp

Types = 3 Tokens = 12
Times used = 3 Times used * Frequency = 12

Semantics:

```
        IF ( IMMED ) ⊆  AUXFCN( [linkp] , [qadp] )
            THEN TRUE ELSE FALSE
```

(4,42) a -> inter linkp advp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
9	1	inter link adv adv	
5	1	inter link adv	

Types = 2 Tokens = 14
Times used = 2 Times used * Frequency = 14

Semantics: [inter] ∩

MEASURE(<linkp,ADVP>,

AUXFCN([linkp] , IMMED) , [advp])

Some utterances using (4,42):

(From: 9 inter#aux,inter#link prep,adv adv)

5	what's in there?
2	what's under there?
1	what's in here?
1	what's out there?

(Remark: Dictionary problems.)

(From: 4 inter#aux,inter#link adv)

3	who's here?
1	what's there?

(4,43) a -> subj vp auxilp

Types = 4 Tokens = 10
Times used = 5 Times used * Frequency = 11

Semantics:

IF ([subj]) ⊆ AUXFCN([auxilp] , [vp])
THEN TRUE ELSE FALSE

(4.44) a -> inter auxilp np verb

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
12	1	inter aux persp v	
10	1	inter aux pron v	
4	1	conj inter aux persp v	
2	1	inter mod persp v	
1	1	conj inter aux pron v	
1	1	inter aux qu n v	

Types = 6 Tokens = 30
Times used = 6 Times used * Frequency = 30

Semantics: [inter] ∩

{ a | (∃ <a,b> ∈

AUXFCN([auxilp] , [verb]))

(b ∈ [np]) } ∩ IMMED

(4.45) a -> subj linkp

Types = 11 Tokens = 32
Times used = 11 Times used * Frequency = 32

Semantics:

IF ([subj]) ⊆ AUXFCN([linkp] , IMMED)
THEN TRUE ELSE FALSE

Some examples:

(From: 7 persp link#aux)

4 i am.
1 it is.
1 we are.

Here the necessity of the contextual parameter IMMED is clear: 'i am' is (probably) not a declaration of existence, but rather asserts that 'i' has some property or another. Again, I feel that having several contextual parameters available will make a needed distinction here.

11. PREPOSITIONAL PHRASE GENERATION

(12.1) prepp -> prep np

Types = 236 Tokens = 479
Times used = 319 Times used * Frequency = 603

Semantics: $\{ a \mid (\exists \langle a,b \rangle \in [prep])(b \in [np]) \}$

12. SUBJECTS OF SENTENCES.

The subj rules generate subjects. No new semantic content is contained in these rules.

(6.1) subj -> np

Types = 823 Tokens = 3342
Times used = 883 Times used * Frequency = 3444

Semantics: [np]

(6,2) subj -> np prepp

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
4	2	v persp prep art n	
2	2	v persp prep n	
2	2	v pron prep persp	
1	2	aux n prep art n	
1	2	aux pron prep art n	
1	2	conj art n prep persp v art n	
1	2	conj mod qu n prep n v neg persp	
1	2	intadv aux art n prep art n	
1	2	inter pron prep art n	
1	2	inter link pron prep pronadj n	
1	1	pronadj n conj pronadj n prep persp v	
1	2	v art n prep pron	
1	2	v n prep n	
1	2	v n prep persp	
1	2	v persp prep persp	
1	3	v persp prep n prep art n n	2
1	2	v pron prep art n n	
1	2	v pronadj n prep pronadj n	
1	2	v qu n prep art n	

Types = 19 Tokens = 24
Times used = 20 Times used * frequency = 25

Semantics: $[np] \cap [prepp]$

Notice that all but one of the forms using (6,2) are grammatically ambiguous. This is because the rule is not really necessary, except for the form

1 pronadj n conj pronadj n prep persp v

where no alternative derivation exists. Semantically, there is no problem since the ambiguity does not affect the semantics. See Section 2 for a discussion of ambiguity.

Some examples of utterances using (6,2):

```
(From:   4       v persp prep art n)

2        put it on the microphone.
1        thank you for a daddy.
1        thank you for a dinner.
```

The intended interpretation of the semantics for (6,2) is that the prepositional phrase modifies the noun phrase. This is usually not the case, so the rule is incorrect.

13. UTTERANCE-GENERATING RULES

The symbol 's' is the start symbol of the grammar GE1.

(8,1) s -> a

```
Types = 836     Tokens = 5037
Times used = 914           Times used * Frequency = 5162

Semantics:        [a]
```

(8,2) s -> afr int

```
Types = 1     Tokens = 541
Times used = 1    Times used * Frequency = 541

Semantics:        TRUE
```

The original utterances for rule (8,2) are:

```
532     uh huh.
  8     uh hum.
```

1 ummm eek.

Clearly, these phrases should be reclassed in the dictionary.

Having a single rule in the grammar to account for these costs nothing, but it doesn't prove anything either. Rule (8,2) simply says that these sentences are grammatical.

(8,4) s -> neg a

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
18	1	neg n	
8	1	neg persp aux neg	
6	1	neg adj	
6	1	neg pron link art n	
5	1	neg art n	
4	1	neg mod persp v persp	
4	1	neg n n	
4	1	neg persp mod neg	
3	1	neg adv	
3	1	neg pron	
3	1	neg v	
2	1	neg adj adj	
2	1	neg pron link n	
2	1	neg persp link n	
2	1	neg persp link art n	
2	1	neg pron link art adj n	
2	1	neg persp mod neg v prep	
2	1	neg qu n	
1	1	neg adj n	
1	1	neg adv adj	
1	1	neg art n conj art n	
1	2	neg mod persp	2
1	1	neg mod persp v pron	
1	1	neg mod persp v prep persp n	
1	1	neg n v	

```
1       1       neg n pn
1       1       neg n mod neg
1       1       neg n pn v prep pronadj n
1       1       neg pn
1       1       neg persp v n
1       1       neg prep qu n
1       1       neg pron link
1       1       neg prep persp
1       1       neg prep padj n
1       1       neg persp v pron
1       1       neg persp v parsp
1       1       neg persp v art n
1       1       neg persp v adj n
1       1       neg persp link pn
1       1       neg prep pronadj n
1       1       neg persp link adj
1       1       neg pron conj pron
1       1       neg pron link pn n
1       1       neg pronadj n aux v
1       1       neg persp aux neg v
1       1       neg persp mod v persp
1       1       neg pron link pronadj
1       1       neg persp v persp pron
1       1       neg persp link neg adj
1       1       neg persp link art pn n
1       1       neg pron link neg art pn
1       1       neg persp link art adj n
1       1       neg persp mod negzv pron
1       1       neg persp aux v prep persp
1       1       neg pron link pronadj adj n
1       1       neg persp mod v prep pronadj n
1       2       neg persp mod neg v pron prep pron       2
1       1       neg v n
1       1       neg v pron
1       1       neg v persp prep
Types = 60    Tokens = 120
Times used = 62   Times used * Frequency = 122
```

Semantics: < FALSE , [a] >

The semantics for this rule is based on the assumption that the utterance is first a negating word (expressing a "complete thought"), followed by a complete sentence. The sentence often explains or elaborates upon the negating word.

For example, the form

6 neg pron link art n

represents the utterances

2 no, that's a butterfly.
no, that's a boy.
no, that's a bear.
no, that's a clock.
no, that's a ocean.

Such utterances must, I believe, be given paired denotations in order to be sensible.

(8,5) s -> aff a

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
11	1	aff persp v	
9	1	aff persp mod	
5	1	aff persp link	
1	1	aff mod persp n	
1	1	aff n	
1	1	aff pron link	
1	1	aff persp link adj	
1	1	aff persp link art n	
1	1	aff persp mod v persp	
1	1	aff prep n prep persp	

Types = 10 Tokens = 32
Times used = 10 Times used * Frequency = 32

Semantics: < TRUE, [a] >

Rule (8,5) and (8,6) which follows have paired denotations for their semantics. Some utterances using (8,5):

(From: 9 aff persp mod)

3	yes, you can.
1	ok, i will.
1	yes you will.
1	yes, i can.
1	yes, he can.
1	yes, it might.
1	yes, she would.

(8,6) s -> a aff

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
1	1	persp mod neg v n aff	
1	1	v aff	

Types = 2 Tokens = 2

Times used = 2 Times used * Frequency = 2

Semantics: < TRUE, [a] >

(8,7) s -> neg

TERMINAL FORMS

Types	No. of Derivations	Form	Times rule used on form (If different from 1)
364	1	neg	

Types = 1 Tokens = 364

Times used = 1 Times used * Frequency = 364

Semantics: FALSE

All 364 of the uses of rule (8,7) represent

364	no.

(8,8) s -> aff

Types = 1 Tokens = 358
Times used = 1 Times used * Frequency = 358

Semantics: TRUE

Utterances involved:

92	uhuh.
66	ok.
59	uh.
41	yeah.
40	yes.
13	yep.
7	yeh.
6	umm.
2	ummm.
	uhmmm.
	uhhmmmm.

The proliferation of these words is not particularly useful for semantics research. It is likely that the editor meant to indicate different pronunciations.

(8,9) s -> int

Types = 1 Tokens = 240
Times used = 1 Times used * Frequency = 240

Semantics: 0

The semantics for an interjection is here considered to be nothing--the empty set. Some examples follow:

92	oh.
44	umhum.

(Remark: 'umhum' is probably an affirmative word.)

10	um.

(Remark: 'um' is probably an affirmative also.)

9	hi.

(8,10) s -> conj

Types = 1 Tokens = 4
Times used = 1 Times used * Frequency = 4

Semantics: 0

These are probably fragments. The utterances using (8,10) are:

2	and...
1	but...
1	even...

(8,11) s -> aff aff

Types = 1 Tokens = 42
Times used = 1 Times used * Frequency = 42

Semantics: < TRUE, TRUE >

The purpose of this rule was to capture two affirmations in one utterance. The original utterances are:

41 uh uh.
1 yeah...yeah.

'uh uh' is clearly just one word. 'yeah...yeah' could conceivably be two separate statements, but the context rules this out. Hence, this rule tries to capture a distinction that simply isn't present in ERICA.

(8.12) s -> int int

Types = 1 Tokens = 59
Times used = 1 Times used * Frequency = 59

Semantics: 0

Again, these utterances are to have no meaning. Some examples:

32 um hum.

(Remark: probably an affirmative word.)

10 oh, oh.
3 um um.
2 oh, darnit.

(8,15) s -> neg neg

Types = 1   Tokens = 5
Times used = 1   Times used * Frequency = 5

Semantics: < FALSE, FALSE >

The semantics for (8,15) is another paired denotation. The utterances involved are:

4	no, no.
1	nope, no.

These are most likely repetitions for emphasis rather than examples of paired denotations.

Rules (8,16) through (8,19) allow an interjection or conjunction to be added before/after utterances without changing the meaning. Notice that these are not recursive rules—i.e., only one such word can be added.

(8,16) s -> conj a

Types = 88   Tokens = 146
Times used = 91   Times used * Frequency = 149

Semantics: [a]

(8,17) s -> a conj

Types = 2   Tokens = 2
Times used = 2   Times used * Frequency= 2

Semantics: [a]

(8,18) s -> int a

Types = 46   Tokens = 81
Times used = 47   Times used * Frequency = 82

Semantics: [a]

(8,19) s -> a int

Types = 8   Tokens = 13
Times used = 8   Times used * Frequency = 13

Semantics: [a]

II. GRAMMATICAL AND SEMANTICAL AMBIGUITY

Chapter 4 contains an extensive discussion of lexical and grammatical ambiguity in the ERICA corpus. That discussion contains the beginning of a discussion of the correctness of the disambiguation. However, correctness of a syntactical construction is a problem that really relates to the intended semantics of the grammar. Hence, I have delayed the consideration of that problem until this time.

I shall consider only the grammatical ambiguity remaining in the ERICA corpus after lexical disambiguation with the probabilistic method. There is relatively little such ambiguity remaining, as shown in Table 1.

TABLE 1

GRAMMATICAL AMBIGUITY IN ERICA

AFTER LEXICAL DISAMBIGUATION

NUMBER OF TREES PER UTTERANCE	TYPES	TOKENS
1	980	6919
2	78	125
3	1	1
4	0	0
5	1	1

Hence, only 80 forms representing 127 utterances have any grammatical ambiguity (using the probabilistic model of lexical disambiguation, which removes some grammatica ambiguity).

I shall say that an utterance κ in sample S is semantically ambiguous iff there are two denotations d1, d2 for κ in some model $\mathfrak{A}$, such that

$$d1_{\mathfrak{A}} \neq d2_{\mathfrak{A}} .$$

Clearly, a terminal form must be grammatically ambiguous in order to be semantically ambiguous (since each production in the grammar concerned has only one associated semantical rule, and since the rules apply in a unique way to a given tree). However, it is clearly possible to have an utterance that is grammatically ambiguous but not semantically ambiguous. An example in ERICA concerns rule

(6,2) subj -> np prepp

(see Section 1). All but one of the forms using (6,2) are grammatically ambiguous. Nevertheless, it is easy to show that there is no semantical ambiguity generated. The form

4 v persp prep art n

uses this rule; the two trees involved are shown in Table 2. Both trees have the denotation:

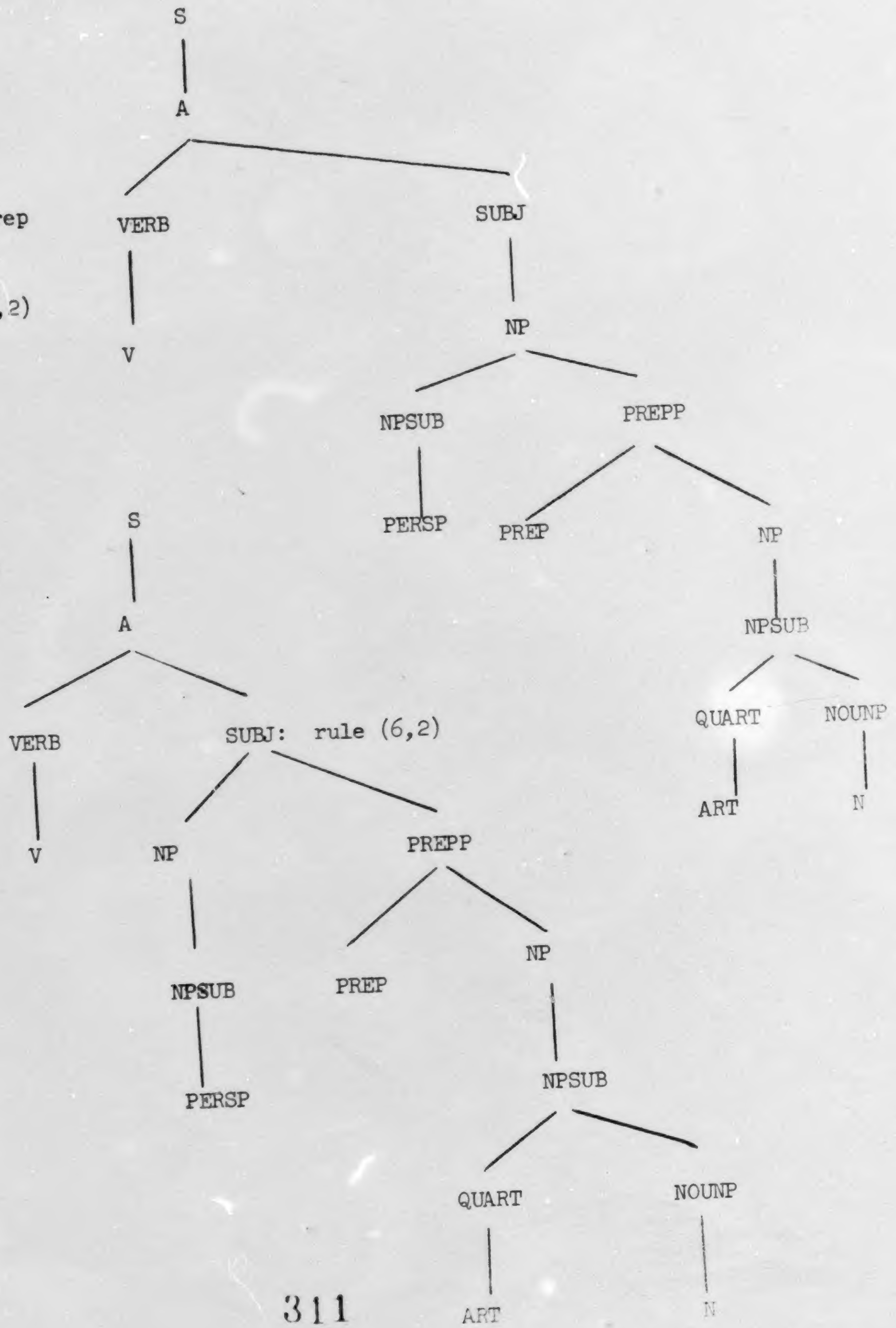
$$\begin{array}{l} \text{if } ([\text{persp}] \cap \\ \quad \{ a \mid (\exists \langle a,b\rangle \in [\text{prep}]) \\ \qquad (b \in \text{QUANTIF}([\text{art}],[\text{n}])) \}) \\ \quad \subseteq [\text{v}] \text{ then TRUE else FALSE} \end{array}$$

TABLE 2

TREE for
'v persp prep
art n'

Without (6,2)

With (6,2)


S
A
VERB
SUBJ
V
NP
NPSUB
PREPP
PERSP
PREP
NP
NPSUB
QUART
NOUNP
ART
N
S
A
VERB
SUBJ: rule (6,2)
V
NP
PREPP
NPSUB
PREP
NP
PERSP
NPSUB
QUART
NOUNP
ART
N

Looking at the original listing of lexical forms (before lexical disambiguation) we find 103 types, representing 137 tokens, that have some grammatical ambiguity. This grammatical ambiguity is traceable to four basic causes in the grammar. These causes of grammatical ambiguity are discussed below, and summarized in Table 3.

1) Prepositional phrase: Does a prepositional phrase modify the noun phrase preceding it (see rule (13,1)) or is it an indirect object of the verb (see rule (3,6))? See Table 4 for the alternative semantic trees for the form

7 persp v,aux qu,pron prep qu,pron.

2) Rule (4,7): The 4 forms using (4,7) are all semantically ambiguous. For example,

5 mod persp

has the semantic trees shown in Table 5. The (syntactically unnecessary) duplication of derivations was originally due to my feeling that some of the utterances involved might require reference to a contextual parameter (IMMED), and others might not require such context checking. As I have examined the many other problems present in the corpus, this one seems irrelevant. I mention it only to show that the technique for giving alternative semantics for a construction is to define separate rules with separate functions.

3) Rule (6,2): As mentioned above, most of the utterances using (6,2) are grammatically ambiguous. However, (6,2) does not create any semantic ambiguity.

4) Adverbial Phrases: Two or more adverbs together cause a semantic ambiguity (see Rules (1,3) and (14,2)). Table 6 has the trees for 'pron qu,pron link,aux adv adv adj'.

This ambiguity is easy enough to eliminate from GE1 once one decides which interpretation to accept. I have allowed it to remain because it illustrates two viable alternative interpretations for adverbial phrases.

5) Rule (4,7) and (6,2) together: Two utterances introduce grammatical ambiguity by using both of these rules together. No other complex causes of grammatical ambiguity are to be found in ERICA.

TABLE 3

CAUSES OF GRAMMATICAL AMBIGUITY IN GRAMMAR GE1

AMBIGUITY	TYPES	TOKENS
PREPOSITIONAL PHRASES	69	89
RULE (4,7)	7	17
RULE (6,2)	19	23
ADVERBIAL PHRASES	6	6
RULES (4,7), (6,2)	2	2

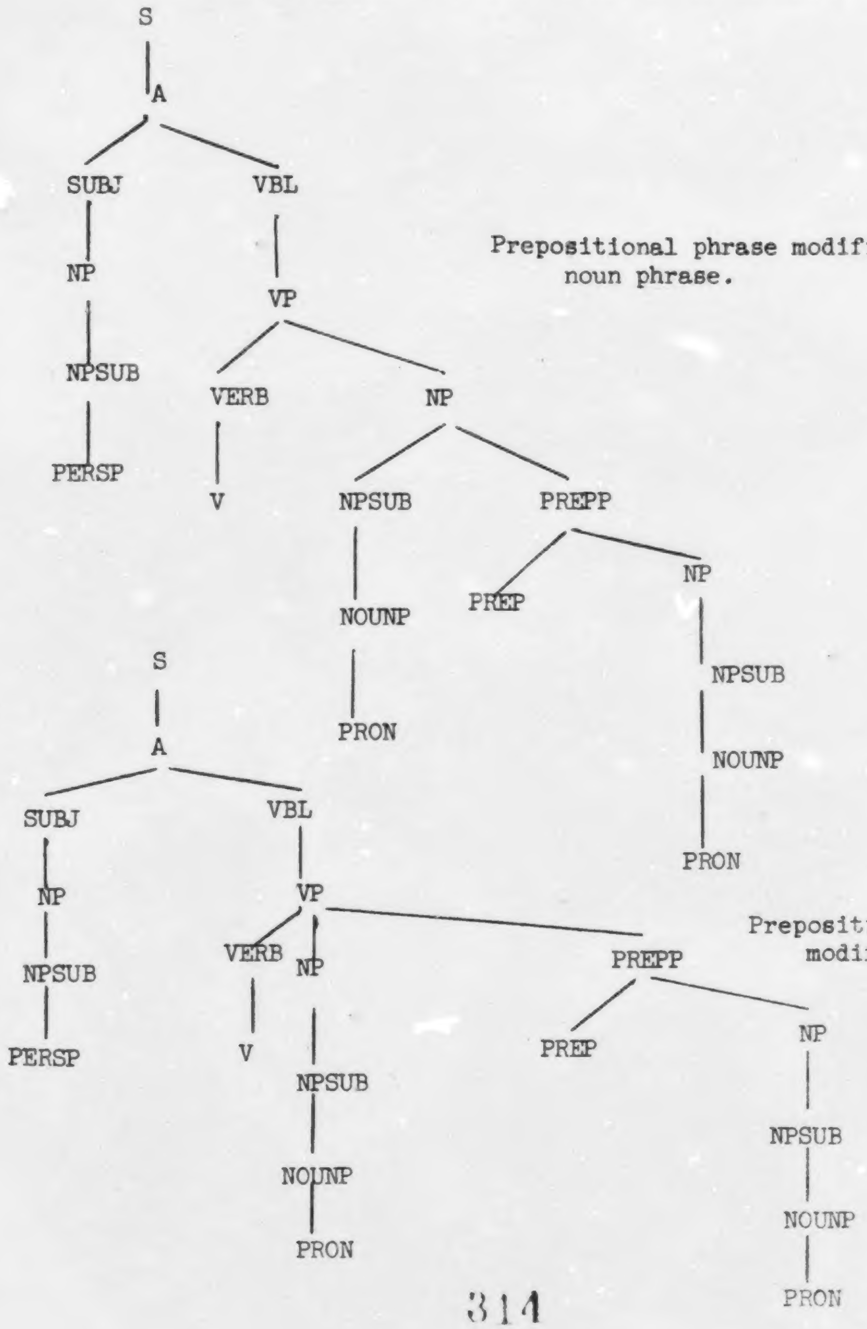
TYPES = 103 TOKENS = 137

TABLE 4

TREES FOR 'PERSP V,AUX QU,PRON PREP QU,PRON'

(Disambiguated as 'persp v pron prep pron'.

The other alternative forms have no derivations.)

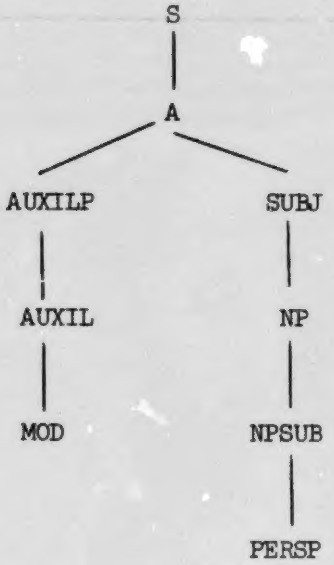

S
A
SUBJ
VBL
NP
VP
NPSUB
VERB
NP
PERSP
V
NPSUB
PREPP
NOUNP
PREP
NP
PRON
NPSUB
NOUNP
PRON
S
A
SUBJ
VBL
NP
VP
NPSUB
VERB
NP
PREPP
PERSP
V
NPSUB
PREP
NP
NOUNP
NPSUB
PRON
NOUNP
PRON

Prepositional phrase modifies noun phrase.

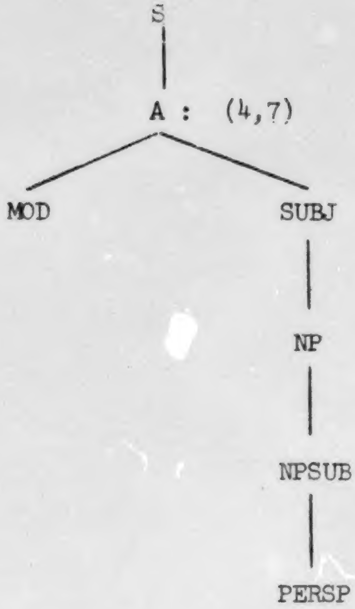
Prepositional phrase modifies verb.

TABLE 5

TREES FOR 'MOD PERSP'


S
A
AUXILP
SUBJ
AUXIL
NP
MOD
NPSUB
PERSP

Without (4,7)


S
A : (4,7)
MOD
SUBJ
NP
NPSUB
PERSP

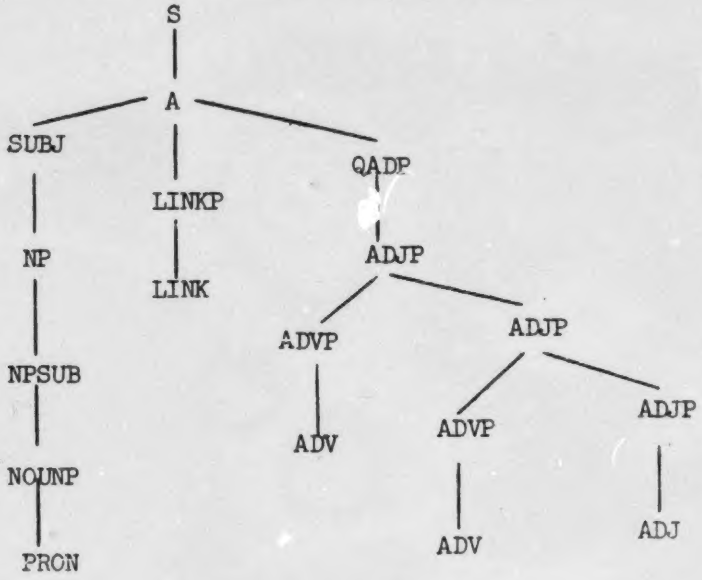
With (4,7)

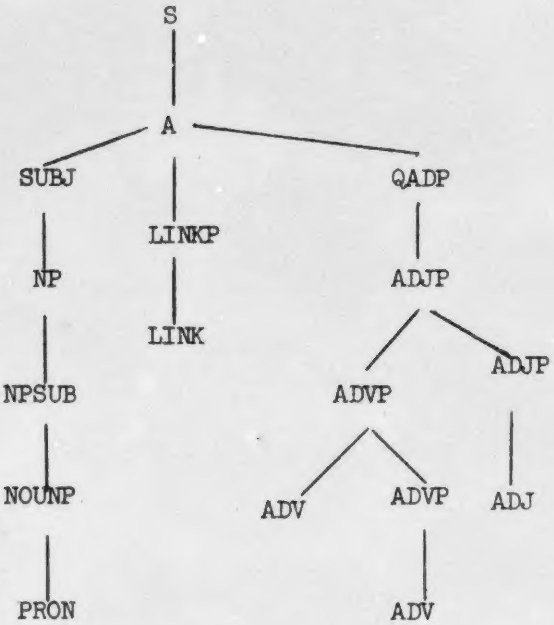
TABLE 6

TREES FOR 'OU,PRON LINK,AUX ADV ADV ADJ'

(The only lexical alternative recognized by GE1 is

'pron link adv adv adj'.)


S
A
SUBJ
LINKP
QADP
NP
LINK
ADJP
NPSUB
ADVP
ADJP
ADV
ADVP
ADJP
NOUNP
ADV
ADJ
PRON


S
A
SUBJ
LINKP
QADP
NP
LINK
ADJP
NPSUB
ADVP
ADJP
NOUNP
ADV
ADVP
ADJ
PRON
ADV

III. PROBABILISTIC DISAMBIGUATION

The major grammatical ambiguity occurring in GE1 is the disposition of the prepositional phrase: is it an indirect object, or does it modify a noun-phrase? The probabilistic grammar obtained by using the values from the probabilistic model of lexical disambiguation (see Chapter 4) assigns a probability of .79 to the indirect object, and .21 to the noun-phrase modifier role.

Examination of the 89 utterances in the listing prior to lexical disambiguation yields the following:

1) Only 21 utterances are (strictly interpreted) indirect objects. Some examples are:

1 i loan it to her.
1 he didn't buy any loaf for him.
1 i gonna share it with you.

GE1 predicts that we would find 71 utterances of this class.

2) A larger than expected 32 utterances have the prepositional phrase modifying the noun. Some examples are:

1 i want one of those.
1 snoopy dog don't have some of that.

(Remark: Most of these utterances have prepositional phrases like 'of these', 'of that', i.e., where the object of the preposition is a 'pron'. GE1 had predicted that we would find only 18 utterances of this kind.)

3) In addition, 36 utterances are adverbial phrases modifying the verb in the utterances. Some examples are:

1 can you see them in the hole?
1 lemme have one in the store.
1 daddy put a fire on it.
1 man fixed my toe on a bed.
1 i go way in the air.
1 i can save them for my room.

GE1 does not consider these adverbial uses of the prepositional phrase.

In several of these utterances the prepositional phrases seem to be objects of the verb. Notice particularly

1 daddy put a fire on it.
1 i can save them for my room.

I think it is clear that the structure of verbs needs to be reconsidered here. Verbs should be classed according to the number of objects expected of them and the rules written to account for different verb symbols. This should also simplify the structure of interrogative adverbs. For example, suppose that the structure of the verb 'go' is

<subject, place>

i.e., the 'place' is where the subject is going to. Then, we would have

[where are you going?] =

$$\{\ b \mid (\exists \langle a,b\rangle \in [\text{are going}])\ (a \in [\text{you}])\ \}$$

This concludes my discussion of the semantics of ERICA.

BIBLIOGRAPHY*

Ajudukiewicz, K. *Jezyki Poznanie*. Warsaw, 1960.

Chomsky, N. Quine's empirical assumptions. In D. Davidson and J. Hintikka (Eds.), *Words and objections: Essays on the work of W. V. Quine*, Dordrecht, Holland: Reidel, 1969. Pp. 53-86.

Gammon, E. M. A syntactic analysis of some first-grade readers. Technical Report No. 155, June 22, 1971, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Hopcroft, J., and Ullman, J. *Formal languages and their relation to automata*. Reading, Mass.: Addison Wesley, 1969.

Kucera, H., and Francis, W. N. *Computational analysis of present day American English*. Providence, R. I.: Brown University Press, 1967.

Montague, R. On the nature of certain philosophical entities. *The Monist*, 1969, *53*, 161-194.

---

* Works listed in the Bibliography are cited in the text by using the bracket convention. Thus, the second listed work authored by P. Suppes is referred to as [Suppes-2].

Montague, R. English as a formal language. In B. Visentini (Ed.), Linguaggi nella societa e technica. Milan, 1970. (a)

Montague, R. Pragmatics and intensional logic. Synthese, 1970, 22, 68-94. (b)

Montague, R. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes (Eds.), Approaches to natural language. Dordrecht, Holland: Reidel, forthcoming.

Suppes, Patrick. Probabilistic grammars for natural languages. Technical Report No. 154, May 15, 1970, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Suppes, Patrick. Semantics of context-free fragments of natural languages. Technical Report No. 171, March 30, 1971, Stanford University, Institute for Mathematical Studies in the Social Sciences.

Tarski, Alfred. The concept of truth in formalized languages. In Logic, Semantics, and Metamathematics. London: Oxford, 1955.

INDEX

ITEM PAGE(S)

alternative terminal form . . 42
ambiguity factor 110
ambiguous lexical form . . . 42
attributivity 170
automata 62

basis valuation 166

chi-square 35, 90, 318
Chomsky normal form 66
clean section 183
closure 132
coefficient of variation . . 313
consolidation 113
context-free 61
context-free semantics . . . 148
context-sensitive 61
contextual ordering 180
contextual parameter 16, 181
corrected observed 102
correction for continuity . . 92

definite description 179
degrees of freedom 90
demonstrative 178
denoting symbol 135
derivable 58
derivation 58
domain 131

equal weights method 88
essential object 187
expected frequency 85

first-order language 131
full parameter model 92

generative grammar 56
geometric 93

geometric distribution . . . 318, 319
goodness of fit 318
grammatical ambiguity 11
grammatically ambiguous . . . 65, 74

imitation 32
immediately produced 57
independent parameter 87
intensive 178

label 65
left-hand side 57
left-most derivation 60
length 23, 61
lexical ambiguity 11
lexical form 67, 94
lexical simplification 67
lexical symbol 67
lexically ambiguous 40, 94
likelihood equation 86
logical form 179
logical symbol 136

maximum likelihood 85
maximum-likelihood 318
mean length of utterance . . 313
modal logic 157
mode 314
modified chi-square 92

n-imitation 32
natural language 18
negative binomial distribution 318, 321
non-uniform function 167
nonterminal vocabulary . . . 56

ontological commitment . . . 155

parameter 72
Pearson's skew statistic . . 314
poisson distribution 318, 320
potentially denoting 159
probability of a tree 72
production 56

recognize 62
recursively-enumerable . . . 61
reduce 110
reduced lexical form 110
regular 61

relational structure 134
relative ambiguity 87
residual 121
resolve 108
resolved lexical form 109
right-hand side 57
right-most derivation 61
rule 56
RULE (1,1) 200
RULE (1,2) 200
RULE (1,3) 201
RULE (10,1) 213
RULE (10,2) 214
RULE (11,1) 251
RULE (11,2) 251
RULE (12,1) 290
RULE (13,1) 215
RULE (13,3) 218
RULE (13,4) 219
RULE (14,1) 205
RULE (14,2) 205
RULE (15,1) 231
RULE (15,2) 231
RULE (16,1) 228
RULE (16,2) 228
RULE (17,1) 220
RULE (17,2) 220
RULE (17,3) 220
RULE (17,4) 220
RULE (17,5) 221
RULE (18,1) 257
RULE (18,2) 257
RULE (19,1) 253
RULE (19,2) 253
RULE (2,1) 214
RULE (2,2) 215
RULE (2,3) 215
RULE (21,1) 207
RULE (21,2) 207
RULE (22,1) 210
RULE (22,2) 210
RULE (22,3) 211
RULE (22,4) 212
RULE (22,5) 213
RULE (3,1) 236
RULE (3,2) 236
RULE (3,3) 238
RULE (3,4) 238
RULE (3,5) 243
RULE (3,6) 245

RULE (3,8) 247
RULE (3,9) 248
RULE (4,1) 257
RULE (4,10) 267
RULE (4,11) 269
RULE (4,12) 269
RULE (4,13) 270
RULE (4,14) 270
RULE (4,15) 270
RULE (4,16) 272
RULE (4,19) 273
RULE (4,2) 261
RULE (4,20) 273
RULE (4,21) 275
RULE (4,22) 276
RULE (4,23) 277
RULE (4,24) 277
RULE (4,25) 278
RULE (4,28) 279
RULE (4,29) 280
RULE (4,3) 262
RULE (4,30) 281
RULE (4,31) 281
RULE (4,32) 282
RULE (4,33) 283
RULE (4,35) 284
RULE (4,37) 284
RULE (4,38) 285
RULE (4,39) 286
RULE (4,4) 262
RULE (4,40) 287
RULE (4,41) 287
RULE (4,42) 288
RULE (4,43) 288
RULE (4,44) 289
RULE (4,45) 289
RULE (4,5) 263
RULE (4,6) 265
RULE (4,7) 265
RULE (4,8) 265
RULE (4,9) 266
RULE (5,1) 222
RULE (5,2) 228
RULE (6,1) 290
RULE (6,2) 291
RULE (7,1) 254
RULE (7,3) 255
RULE (7,4) 256
RULE (7,5) 256
RULE (8,1) 292

RULE (8,10) 298
RULE (8,11) 298
RULE (8,12) 299
RULE (8,15) 300
RULE (8,16) 300
RULE (8,17) 300
RULE (8,18) 301
RULE (8,19) 301
RULE (8,2) 292
RULE (8,4) 293
RULE (8,5) 295
RULE (8,6) 296
RULE (8,7) 296
RULE (8,8) 297
RULE (8,9) 297
RULE (9,1) 208
RULE (9,2) 208
RULE (9,3) 208
rule class 65

sample 85
semantic ambiguity 11
semantical rule 134
semantically ambiguous . . . 160
sentence type 94
simple closure 137
standard deviation 313
start symbol 56
string 56

terminal form 43, 94
terminal vocabulary 56
theoretical frequency 85
tree 63
truncated geometric 93
type-0 61
type-1 61
type-2 61
type-3 61

uniform model 166
USAGE 87
utility symbol 136

valuation 134
variance 313
vocabulary 56

word 23

Yule's K-factor 314

(Continued from inside front cover)

96 R. C. Atkinson, J. W. Breisford, and R. M. Shiffrin. Multi-process models for memory with applications to a continuous presentation task. April 13, 1966. (*J. math. Psychol.*, 1967, *4*, 277-300).

97 P. Suppes and E. Crothers. Some remarks on stimulus-response theories of language learning. June 12, 1966.

98 R. Bjork. All-or-none subprocesses in the learning of complex sequences. (*J. math. Psychol.*, 1968, *1*, 182-195).

99 E. Gammon. The statistical determination of linguistic units. July 1, 1966.

100 P. Suppes, L. Hyman, and M. Jerman. Linear structural models for response and latency performance in arithmetic. (In J. P. Hill (ed.), *Minnesota Symposia on Child Psychology*. Minneapolis, Minn.: 1967. Pp. 160-200).

101 J. L. Young. Effects of intervals between reinforcements and test trials in paired-associate learning. August 1, 1966.

102 H. A. Wilson. An investigation of linguistic unit size in memory processes. August 3, 1966.

103 J. T. Townsend. Choice behavior in a cued-recognition task. August 8, 1966.

104 W. H. Batchelder. A mathematical analysis of multi-level verbal learning. August 9, 1966.

105 H. A. Taylor. The observing response in a cued psychophysical task. August 10, 1966.

106 R. A. Bjork. Learning and short-term retention of paired associates in relation to specific sequences of interpresentation intervals. August 11, 1966.

107 R. C. Atkinson and R. M. Shiffrin. Some Two-process models for memory. September 30, 1966.

108 P. Suppes and C. Ihrke. Accelerated program in elementary-school mathematics--the third year. January 30, 1967.

109 P. Suppes and I. Rosenthal-Hill. Concept formation by kindergarten children in a card-sorting task. February 27, 1967.

110 R. C. Atkinson and R. M. Shiffrin. Human memory: a proposed system and its control processes. March 21, 1967.

111 Theodore S. Rodgers. Linguistic considerations in the design of the Stanford computer-based curriculum in initial reading. June 1, 1967.

112 Jack M. Knutson. Spelling drills using a computer-assisted instructional system. June 30, 1967.

113 R. C. Atkinson. Instruction in initial reading under computer control: the Stanford Project. July 14, 1967.

114 J. W. Breisford, Jr. and R. C. Atkinson. Recall of paired-associates as a function of overt and covert rehearsal procedures. July 21, 1967.

115 J. H. Steizer. Some results concerning subjective probability structures with semiorders. August 1, 1967

116 D. E. Rumelhart. The effects of interpresentation intervals on performance in a continuous paired-associate task. August 11, 1967.

117 E. J. Fishman, L. Keller, and R. E. Atkinson. Massed vs. distributed practice in computerized spelling drills. August 18, 1967.

118 G. J. Groen. An investigation of some counting algorithms for simple addition problems. August 21, 1967.

119 H. A. Wilson and R. C. Atkinson. Computer-based instruction in initial reading: a progress report on the Stanford Project. August 25, 1967.

120 F. S. Roberts and P. Suppes. Some problems in the geometry of visual perception. August 31, 1967. (*Synthese*, 1967, *17*, 173-201)

121 D. Jamison. Bayesian decisions under total and partial ignorance. D. Jamison and J. Kozielecki. Subjective probabilities under total uncertainty. September 4, 1967.

122 R. C. Atkinson. Computerized instruction and the learning process. September 15, 1967.

123 W. K. Estes. Outline of a theory of punishment. October 1, 1967.

124 T. S. Rodgers. Measuring vocabulary difficulty: An analysis of item variables in learning Russian-English and Japanese-English vocabulary parts. December 18, 1967.

125 W. K. Estes. Reinforcement in human learning. December 20, 1967.

126 G. L. Wolford, D. L. Wessel, W. K. Estes. Further evidence concerning scanning and sampling assumptions of visual detection models. January 31, 1968.

127 R. C. Atkinson and R. M. Shiffrin. Some speculations on storage and retrieval processes in long-term memory. February 2, 1968.

128 John Holmgren. Visual detection with imperfect recognition. March 29, 1968.

129 Lucille B. Mlodnosky. The Frostig and the Bender Gestalt as predictors of reading achievement. April 12, 1968.

130 P. Suppes. Some theoretical models for mathematics learning. April 15, 1968. (*Journal of Research and Development in Education*, 1967, *1*, 5-22)

131 G. M. Olson. Learning and retention in a continuous recognition task. May 15, 1968.

132 Ruth Norene Hartley. An investigation of list types and cues to facilitate initial reading vocabulary acquisition. May 29, 1968.

133 P. Suppes. Stimulus-response theory of finite automata. June 19, 1968.

134 N. Moler and P. Suppes. Quantifier-free axioms for constructive plane geometry. June 20, 1968. (In J. C. H. Gerretsen and F. Oort (Eds.), *Compositio Mathematica*. Vol. 20. Groningen, The Netherlands: Wolters-Noordhoff, 1968. Pp. 143-152.)

135 W. K. Estes and D. P. Horst. Latency as a function of number or response alternatives in paired-associate learning. July 1, 1968.

136 M. Schlag-Rey and P. Suppes. High-order dimensions in concept identification. July 2, 1968. (*Psychom. Sci.*, 1968, *11*, 141-142)

137 R. M. Shiffrin. Search and retrieval processes in long-term memory. August 15, 1968.

138 R. D. Freund, G. R. Loftus, and R.C. Atkinson. Applications of multiprocess models for memory to continuous recognition tasks. December 18, 1968.

139 R. C. Atkinson. Information delay in human learning. December 18, 1968.

140* R. C. Atkinson, J. E. Holmgren, and J. F. Juola. Processing time as influenced by the number of elements in the visual display. March 14, 1969.

141 P. Suppes, E. F. Loftus, and M. Jerman. Problem-solving on a computer-based teletype. March 25, 1969.

142 P. Suppes and Mona Morningstar. Evaluation of three computer-assisted instruction programs. May 2, 1969.

143 P. Suppes. On the problems of using mathematics in the development of the social sciences. May 12, 1969.

144 Z. Domotor. Probabilistic relational structures and their applications. May 14, 1969.

145 R. C. Atkinson and T. D. Wickens. Human memory and the concept of reinforcement. May 20, 1969.

146 R. J. Titiev. Some model-theoretic results in measurement theory. May 22, 1969.

147 P. Suppes. Measurement: Problems of theory and application. June 12, 1969.

148 P. Suppes and C. Ihrke. Accelerated program in elementary-school mathematics--the fourth year. August 7, 1969.

149 D. Rundus and R.C. Atkinson. Rehearsal in free recall: A procedure for direct observation. August 12, 1969.

150 P. Suppes and S. Feldman. Young children's comprehension of logical connectives. October 15, 1969.

(Continued on back cover)

(Continued from inside back cover)

151 Joaquim H. Laubsch. An adaptive teaching system for optimal item allocation. November 14, 1969.

152 Roberta L. Klatzky and Richard C. Atkinson. Memory scans based on alternative test stimulus representations. November 25, 1969.

153 John E. Holmgren. Response latency as an indicant of information processing in visual search tasks. March 16, 1970.

154 Patrick Suppes. Probabilistic grammars for natural languages. May 15, 1970.

155 E. Gammon. A syntactical analysis of some first-grade readers. June 22, 1970.

156 Kenneth N. Wexler. An automaton analysis of the learning of a miniature system of Japanese. July 24, 1970.

157 R. C. Atkinson and J.A. Paulson. An approach to the psychology of instruction. August 14, 1970.

158 R.C. Atkinson, J.D. Fletcher, H.C. Chetin, and C.M. Stauffer. Instruction in initial reading under computer control: the Stanford project. August 13, 1970.

159 Dewey J. Rundus. An analysis of rehearsal processes in free recall. August 21, 1970.

160 R.L. Klatzky, J.F. Juola, and R.C. Atkinson. Test stimulus representation and experimental context effects in memory scanning.

161 William A. Rottmayer. A formal theory of perception. November 13, 1970.

162 Elizabeth Jane Fishman Loftus. An analysis of the structural variables that determine problem-solving difficulty on a computer-based teletype. December 18, 1970.

163 Joseph A. Van Campen. Towards the automatic generation of programmed foreign-language instructional materials. January 11, 1971.

164 Jamesine Friend and R.C. Atkinson. Computer-assisted instruction in programming: AID. January 25, 1971.

165 Lawrence James Hubert. A formal model for the perceptual processing of geometric configurations. February 19, 1971.

166 J. F. Juola, I.S. Fischler, C.T. Wood, and R.C. Atkinson. Recognition time for information stored in long-term memory.

167 R.L. Klatzky and R.C. Atkinson. Specialization of the cerebral hemispheres in scanning for information in short-term memory.

168 J.D. Fletcher and R.C. Atkinson. An evaluation of the Stanford CAI program in initial reading (grades K through 3). March 12, 1971.

169 James F. Juola and R.C. Atkinson. Memory scanning for words versus categories.

170 Ira S. Fischler and James F. Juola. Effects of repeated tests on recognition time for information in long-term memory.

171 Patrick Suppes. Semantics of context-free fragments of natural languages. March 30, 1971.

172 Jamesine Friend. Instruct coders' manual. May 1, 1971.

173 R.C. Atkinson and R.M. Shiffrin. The control processes of short-term memory. April 19, 1971.

174 Patrick Suppes. Computer-assisted instruction at Stanford. May 19, 1971.

175 D. Jamison, J.D. Fletcher, P. Suppes and R.C. Atkinson. Cost and performance of computer-assisted instruction for compensatory education.

176 Joseph Offir. Some mathematical models of individual differences in learning and performance. June 28, 1971.

177 Richard C. Atkinson and James F. Juola. Factors influencing speed and accuracy of word recognition. August 12, 1971.

178 P. Suppes, A. Goldberg, G. Kanz, B. Searle and C. Stauffer. Teacher's handbook for CAI courses. September 1, 1971.

179 Adele Goldberg. A generalized instructional system for elementary mathematical logic. October 11, 1971.

180 Max Jerman. Instruction in problem solving and an analysis of structural variables that contribute to problem-solving difficulty. November 12, 1971.

181 Patrick Suppes. On the grammar and model-theoretic semantics of children's noun phrases. November 29, 1971.

182 Georg Kreisel. Five notes on the application of proof theory to computer science. December 10, 1971.

183 James Michael Moloney. An investigation of college student performance on a logic curriculum in a computer-assisted instruction setting. January 28, 1972.

184 J.E. Friend, J.D. Fletcher and R.C. Atkinson. Student performance in computer-assisted instruction in programming. May 10, 1972.

185 Robert Lawrence Smith, Jr. The syntax and semantics of erica. June 14, 1972.